Número 15
JULIO 2008



2 EDICION ANIVERSARIO

2° Aniversario Revista Begins

Y estrenamos sitio web :)

Además

- Entendiendo Web Services con MonoDevelop I de IV
- Los Ambientes de escritorio en Linux
- Y mucho más

Aprovechando la consola para descargar con BitTorrent



La seguridad de nuestras comunicaciones.



Linux y la Brecha Digital







Juegos



Editorial

Redacción

Rodrigo Ramírez	decipher@chile.com
Óscar Calle	reygecko@gmail.com
Álex Sandoval	alexsandovalm@yahoo.com
Esteban Saavedra Lopez	jesaavedra@opentelematics.org
Martin Marquez	xomalli@gmail.com
Gabriel Lopez	llopezgabriel@gmail.com
Wilson Castillo	wilxonkastillo@yahoo.es
Luis Álvarez Alday	luisalvarezalday@gmail.com

Revisión y corrección

Éric Báez	ericbaez@gmail.com
Óscar Calle	reygecko@gmail.com
Guillermo Molleda	gmolleda@us.es

Staff y Coordinación

Luis Álvarez Alday	luisalvarezalday@gmail.com
Óscar Calle	reygecko@gmail.com
Dionisio Fernández	zeroblack@blackhole.cl
Álvaro Parra	alverich@gmail.com
Rodrigo Ramírez	decipher@chile.com
Álex Sandoval Morales	alexsandovalm@yahoo.com
Éric Báez Bezama	ericbaez@gmail.com

Arte y Diseño

Álex Sandoval Morales
http://www.microteknologias.cl

Luis Álvarez Alday
http://www.muestrate.cl/

Es un momento de cambios en Begins.

Por un lado, cambios muy agradables, como la incorporación de Guillermo Molleda como miembro estable de la plantilla. Ni os podéis imaginar la energía que ha insuflado a la revista. ;)

Además (¡¡POR FIN!!) ya está en marcha la web de Begins, http://revistabegins.org/ donde pretendemos centralizar y tener un mejor contacto con los lectores, además de ofrecer diverso contenido nuevo.

Y ahora las malas noticias. ¿Quieres que Begins siga existiendo? Porque si es así, necesitamos tu ayuda, y no estamos hablando de ayuda económica (aunque estaría bien XDDD) sino de tu colaboración.

Begins nació como una revista hecha por la comunidad del software libre y para esa misma comunidad. Aquellos que formamos su plantilla, por diversos compromisos laborales, familiares, etc., disponemos de menos tiempo del necesario para mantener el nivel de calidad en sus contenidos que siempre ha caracterizado a la publicación.

Necesitamos brazos nuevos, cerebros nuevos. Ponte en contacto con nosotros si crees que este proyecto merece la pena.

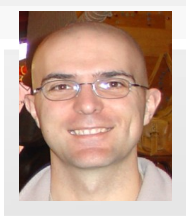
Saludos, linuxeros :D



El nombre y logo de Begins
son invención de Robin Osorio
Adaptación: Luis Alvarez A.

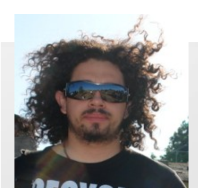


Equipo estable de Begins



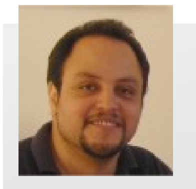
Óscar Calle Gutiérrez
Nick: ReyGecko

Policía en el ayuntamiento de Pinto (Madrid-España).Actualmente está en la carrera de informática en la Universidad Nacional de Educación a Distancia.
Ubicación: Pinto, España.



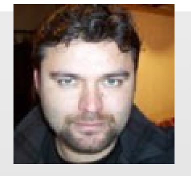
Dionisio Fernández Mora
Nick: ZeRobLaCk

Ingeniero de Ejecución en Computación e Informática. Co-Fundador de LinuxChillan.
Ubicación: Chillán, Chile.



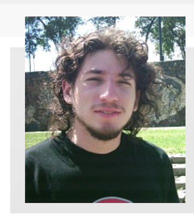
Luis Alvarez Alday
Nick: Papeluis

Webmaster y Diseñador Gráfico amateur, administra el portal www.muestrate.cl
Ubicación: Normalmente en este planeta. Chillán



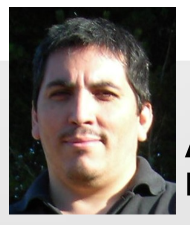
Érick Báez Bezama
Nick: Erick

Periodista, activista social por el libre conocimiento. Socio y fundador del Centro de Difusión de Software Libre (www.cdsl.cl) Editor de www.softwarelibre.cl, integra equipo de traductores de Ubuntu, colaborador de Wikipedia.org
Ubicación: Santiago, Chile



Rodrigo Ramírez Norambuena
Nick: Decipher

Ingeniero de Ejecución en Computación e Informática. Participa en LinuxChillan casi desde sus inicios.
Ubicación: Chillán, Chile.



Alex Sandoval Morales
Nick: DarkSoft

Programador computacional, administra un sitio web para promover Linux. (http://www.microteknologias.cl)
Ubicación: Peñalolén, Chile



Alvaro Parra Urbina
Nick: Alverich

Ingeniero agrónomo en la Universidad de Concepción, Chile.
Ubicación: Chillán, Chile



Guillermo Molleda Jimena
Nick: Profe

Profesor en E.U. de Estudios Empresariales
Ubicación: Valladolid, España

Begins es realizada completamente con Software Libre

Maquetación y empaquetado PDF
OpenOffice

Retoque Fotográfico
The Gimp

Ilustraciones y Diseño Vectorial
Xara Xtreme LE

Diseño 3D
Blender

Capturas de pantalla
KSnapshot



Contenido

Destacado

Artículos

Secciones





LINUX Y LA BRECHA DIGITAL



ENTENDIENDO LA BRECHA DIGITAL.

El concepto de brecha digital se refiere a la distancia que separa a naciones desarrolladas de subdesarrolladas de lo último en tecnología de la información, esto es computación, internet, medios audiovisuales (cámaras digitales, de vídeo, expansión del mercado de videojuegos entre otros)y otros medios tecnológicos que contribuyen al desarrollo social, económico y cultural de un país.

Este concepto se utiliza para atenuar las diferencias tecnológicas en países subdesarrollados, donde se presenta la mayor brecha digital, en especial en países de altos índices de pobreza en África y América Latina, esta brecha en Sudamérica es medida mediante la encuesta Tecnologías de la Información y comunicación T.I.C.

"La brecha digital es un concepto que involucra diversos factores, existe la manera en que Linux pueda ayudar a disminuirla si hay un trabajo adecuado."

LA SITUACION DE AMÉRICA LATINA.

La brecha digital entre América latina y los países desarrollados es alarmante, en naciones de alto poder adquisitivo como Corea, 550 de cada 1.000 personas se conecta a internet a diario, en 10 millones de conexiones de banda ancha para 48 millones de personas, mientras que en Chile, país líder en América latina en cuanto a digitalización, la encuesta TIC publicada en el año 2006 el 36% de la población nacional posee al menos un computador en casa, lo que si bien es cierto es una buena cifra para el sub continente solo el 15% de la población, osea, aproximadamente 2 millones de personas, puede conectarse a internet desde sus hogares, lo que demuestra que solo la población privilegiada, mas una parte de la clase media, puede disfrutar abiertamente de este medio tecnológico.

Estos datos son aun más duros en el resto de la región, ya que el número de conexiones a internet, la velocidad promedio y la cantidad de hogares con al menos un computador en casa son drásticamente inferiores,datos que pueden comprobarse al revisar estadísticas de países como Nicaragua,donde solo el 8% de los consultados posee computador en casa, el 20% se conecta a internet y de estos, el 53% lo hace desde cibercafes, el resto se divide en trabajo, colegio, universidad y en los hogares.

Estas cifras revelan el atraso que la región presenta con respecto al mundo en materia de tecnología.



Todo esto refleja que como país debemos superar muchos aspectos para disminuir la brecha digital,que no solo nos aleja del desarrollo económico, sino también de un crecimiento cultural, educacional y social, y para ello necesitamos de una estrategia a nivel nacional para acortar esta diferencia en la tecnología, que debe abarcar todos los sectores de la sociedad, y es en esta gran estrategia global de acercamiento de la población a la tecnología donde el software libre juega su importante y decisivo papel.

LINUX COMO HERRAMIENTA SOCIAL.

La GPL (General Public License) es una licencia de software que se caracteriza por permitir el copiado y difusión del programa escrito bajo su tutela, por lo que se garantiza que estos son 100% originales, sin limitaciones de ningún tipo, y a costo cero, o bien, a un precio acorde a los bolsillos de la gente común y corriente, que ve al PC como una simple herramienta de trabajo más que como una estación multimedia (que tanto nos quisieron vender como concepto en los años 90).

GNU/LINUX es un sistema operativo escrito como software libre, por lo que su costo como bien se dijo será muy inferior al de un programa del mismo tipo, de origen privativo, lo que lo hace perfecto para equipar computadores de personas de escaso poder adquisitivo que de otro modo no podrían adquirir un computador, lo que permite que hoy más que nunca, muchos puedan acceder a internet y al uso extensivo de computadores desde su casa, de forma 100% legal en un sistema operativo que ha evolucionado de vertiginosa forma en los últimos 4 años en su distribución más popular, Ubuntu.

El sistema operativo basado en Debian Ubuntu, mezcla lo amigable y sencillo de un escritorio digital con el poder del software libre, sólidamente construido por miles de personas que lo siguen mejorando día a día para que, como su lema lo indica, sea usado por personas comunes y corrientes (Linux for humans beigns) lo que acerca la computación y el software libre a personas sin conocimiento alguno en la materia, que es algo que puede transformarse en la mayor oportunidad de nuestro país para dar ese gran salto que tantas veces hemos estado a punto de dar hacia el desarrollo, ya que Ubuntu permitiría que muchos puedan acceder a un sistema operativo a un costo asequible, que se transformaría en una eficiente herramienta de trabajo, estudio, y porque no, diversión, lo que democratizaría por fin la computación e internet en nuestro país, mejorando enormemente un aspecto cada vez mas importante en nuestras vidas como es el acceso a la información.

Así pues, el software libre se une a la sencillez de uso de Ubuntu, que acerca mas que nunca las enormes posibilidades que ofrece el software libre a la gente común, la que no tendrá que elegir entre el software privativo o nada.

UNA ALTERNATIVA LEGAL.

Ubuntu, como todas las distribuciones Linux, esta echa en los términos de la GPL mencionada mas arriba, lo que hace legal copiar y distribuir el software de forma legal, algo que con el pasar del tiempo se esta haciendo mas importante, ya que muchas personas para evitar pagar los altos costos de las licencias privativas recurren a estos programas de forma ilegal, adquiriéndolos ya sea en el mercado negro, internet, o bien, recurriendo a la copia entre amigos, algo prohibido en el mundo del software privativo.
Si se tiene en cuenta que las leyes del país se endurecen cada vez mas al restringir esta distribución de software, llegará el momento en que solo quienes realmente pueden pagar por un sistema operativo privativo podrán optar a el, dejando a un porcentaje importante de población sin alternativa ... o si?

Nuevamente Ubuntu tiene un punto a favor, pues mientras mas se restrinja el pirateo de software, mayor será el mercado disponible para una distribución libre, lo que hace imprescindible difundirla para que la gente tome conciencia y opte por lo legal hoy, y no cuando comiencen a ser castigadas por no haberse cambiado a tiempo a un software que no restringe, sino que integra.

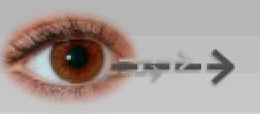
UBUNTU NO ES TODO, TAMBIÉN, LA VOLUNTAD ES IMPORTANTE.

Lógicamente, Ubuntu no es la panacea,si bien es cierto es una ayuda importante en la titánica tarea por democratizar la tecnología, el acceso y uso de esta, no lo es todo, también hace falta un compromiso a nivel de país para lograr superar este desafío.

Todos debemos unirnos en esta tarea. Empresas, gobierno, y la población, que debemos difundir el software libre, acercarlo a quienes no lo conozcan, y de esa forma podremos contar con una alternativa legal que no requiera los abusivos cobros del software privativo para instalarse en nuestros equipos, entregándonos una variada gama de programas que facilitarán nuestra vida estudiantil, laboral y de ocio.

Estamos a las puertas de dar un importante salto, el software libre viene a ayudarnos, y no solo a mejorar el nivel de vida del país acercando la tecnología a todos, también haciéndolo de forma licita, sin fomentar la piratería que hace perder al estado millones de dolares al año en impuestos. Estamos pues, ante la mayor ayuda que nunca antes hayamos podido tener para acercar el mundo digital a los demás, y haciéndolo con un software echo por personas que creen que el dinero no es todo, la solidaridad también puede mover montañas, y lo demuestran día a día con un software de excelente calidad como Ubuntu.

Mientras que el software privativo permite a una empresa ganar millones de dolares al año, el software libre permite a millones de personas ganar en calidad de vida al año, y eso, es algo que no tiene precio.



La seguridad de nuestras comunicaciones.



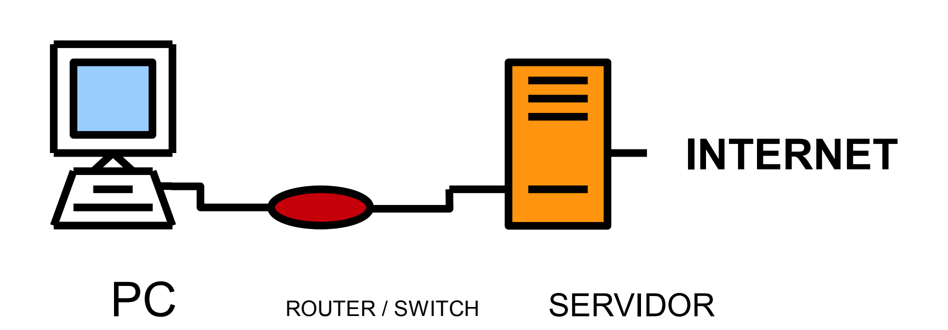
En esta ocasión, vamos a tratar que cualquier usuario novato pueda entender un concepto que, cada día más, tiene una importancia creciente en nuestra relación diaria con Internet: *la seguridad de nuestras comunicaciones. Muchos de nosotros interactuamos de forma habitual con nuestro banco mediante Internet pero... ¿es eso seguro? ¿Deberíamos preocuparnos? ¿Cómo podemos minimizar los riesgos al tiempo que podamos seguir realizando nuestras operaciones bancarias de una forma segura? ¿Son asimismo seguras las compras on-line? Vayamos paso a paso...*

En el mundo de la computación, como en la película "*The Matrix*", hay muy pocas cosas que sean completamente imposibles. Si alguien fue capaz de comprar viagra y cargarlo a la cuenta de nuestro querido Bill Gates, imaginad lo que podría haber hecho con nuestra propia cuenta. Pero eso no impide que podamos tomar en cuenta una serie de medidas que dificulten mucho el trabajo al posible delincuente cibernético, tanto que seguro preferirá buscar otra víctima.

Lo primero que debemos tener en cuenta es la forma en la que nos conectamos a internet. ¿Cómo lo hacemos? ¿Mediante cable o de forma inalámbrica (wifi)? Debemos tener en cuenta que, si lo hacemos de forma inalámbrica, nuestras conexiones podrán ser espiadas por terceras personas que se encuentren próximas a nosotros. Es lo que se conoce como "sniffar" una conexión.

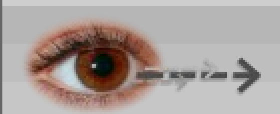
Por lo tanto, la primera medida que deberíamos contemplar es, si nos resulta posible, evitar la conexión inalámbrica y conectarnos por cable. Lo sentimos, sniffers, otra vez será... ;)

Ahora bien, una conexión por cable no significa que sea una conexión segura. ¿Nos conectamos directamente a internet o actuamos "a través de un servidor"? Por ejemplo, si estamos en nuestro puesto de trabajo, en la universidad, etc., nuestra conexión a internet (aunque sea mediante cable) pocas veces será directa sino que lo más probable es que acceda a internet a través de un servidor de la propia empresa, facultad, biblioteca, ciber-café o lo que sea. Algo como esto:



En este caso, como podemos ver, nos estaríamos conectando a internet a través del servidor de la empresa. Por lo tanto, todas nuestras comunicaciones pasarán por ese punto. ¿Querrías que tus datos bancarios, tus contraseñas, etc., pasasen por ahí sabiendo que podrán ser accesibles para los responsables del área de informática de tu empresa?

Aún mas: suponiendo que confías al 100% en esos responsables, ¿quién te asegura que el servidor de tu empresa no está infectado por un keylogger, un troyano o cualquier otro similar?



Por este motivo, lo mejor es conectarse a nuestro banco de forma directa, desde nuestro domicilio. Pero... ¿y si tenemos un “keylogger” instalado en nuestro sistema? Buena pregunta...

En realidad, resulta mucho más complicado que se instale un keylogger o programa similar en nuestro sistema. Por su propia construcción, por su esqueleto, los sistemas Linux son mucho más robustos que los sistemas Windows (que por defecto, hasta la versión XP, viene instalado sin medidas de seguridad), lo que dificulta en buena medida la tarea de los amigos de lo ajeno. Pero eso no es suficiente. Como dije al principio, en computación no hay casi nada imposible y, a pesar de ser un sistema linux, podría estar afectado por algún troyano o similar que enviase información sobre nuestros datos bancarios.

NOTA: un “*keylogger*” es un programa que se ejecuta normalmente en segundo plano (es decir, que no lo ves ejecutándose) y que registra todas las pulsaciones de teclas en el equipo donde se encuentra instalado. Podría, por tanto, recuperar unas contraseñas tecleadas aunque por pantalla sólo se muestren los típicos asteriscos o puntos negros. Del mismo modo, existen “*videologgers*”, aunque por fortuna son muchísimo menos frecuentes.

Entonces ¿qué podemos hacer? Pues hay una solución muy sencilla que nos asegurará casi al 100% que nadie podrá acceder a nuestros datos. Al menos mediante un software tipo troyano o similar, aunque siempre podrían intentar hackear vuestro ISP, vuestro proveedor de internet, pero eso se escapa de nuestras posibilidades preventivas, ya dije que no hay sistema completamente seguro.

Repito la pregunta: ¿qué podemos hacer? De nuevo, Linux viene a rescatarnos. Mi recomendación personal es que visitéis esta página: http://www.sabayonlinux.org

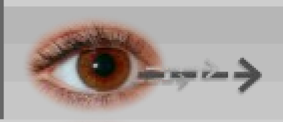
En ella, podemos entrar en la sección de descargas (http://www.sabayonlinux.org/mod/mirrors) y bajarnos la ISO de Sabayon Linux, una excelente distribución basada en Gentoo que nos permitirá (entre otras cosas) disfrutar de los efectos de Compiz Fusion desde el live-dvd. Algunos lugares directos de descarga son estos:



http://mirror.internode.on.net/pub/sabayonlinux/SabayonLinux-x86-3.4f.iso
ftp://na.mirror.garr.it/mirrors/sabayonlinux/SabayonLinux-x86-3.4f.iso
ftp://ftp.sh.cvut.cz/MIRRORS/sabayon/SabayonLinux-x86-3.4f.iso

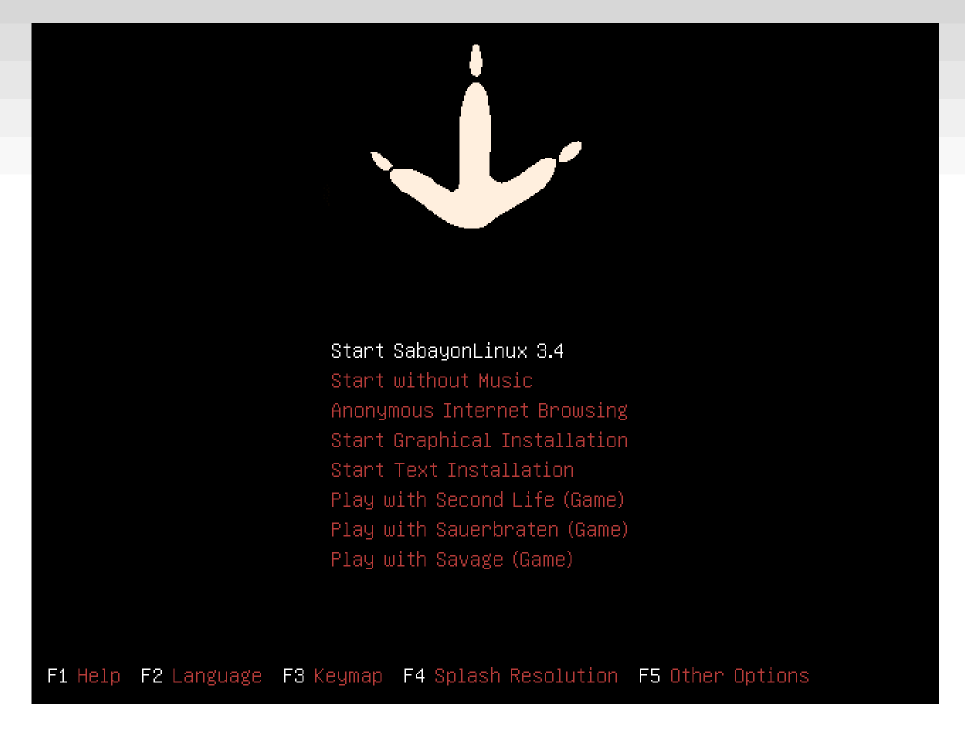
Tras bajar la ISO, podremos grabarla en un DVD y ya tendremos un live-dvd de Sabayon Linux para poder probarlo.





La seguridad de nuestras comunicaciones.

Iniciando nuestro PC desde el live-dvd de Sabayon, veremos un menú donde podremos elegir nuestro idioma pulsando F2, tras lo cual podremos iniciar normalmente.



Alguien se podría preguntar “bien pero... ¿no podemos hacer esto igualmente desde el cd de Ubuntu?”. Dos motivos para no hacerlo:

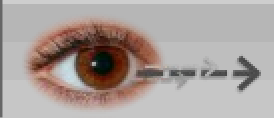
1.- Conoceremos otra distribución de GNU/Linux, basada en un sistema distinto y con un entorno también diferente. Eso nos ayudará a entender que el mundo del pingüino ni empieza ni acaba en Ubuntu. Ampliará nuestra visión de lo que significa un sistema Linux.

2.- Ubuntu no lleva instalado “de serie” ni el plugin de flash ni el de Java (uno o los dos plugins mencionados pueden ser requeridos por nuestra entidad bancaria para poder acceder como usuarios conectados). Es más, en el momento en que se escribe este artículo resulta imposible la instalación del plugin de Java en Ubuntu funcionando desde el live-cd. Respecto del plugin para flash, podría explicar cómo instalarlo en la live-cd de Ubuntu pero preferí no hacerlo por el motivo nº1.

Al conectarnos a nuestro banco desde un live-dvd, no tocaremos nuestro disco duro (funciona desde el dvd, ni siquiera necesita que tu equipo tenga disco duro) por lo que si tuviésemos algún programa “espía” instalado, éste no se activará.

Resumiendo: conectándonos desde nuestra casa con un live-cd evitamos sniffers, troyanos, programas espía y cualquier posible injerencia externa que pudiera derivarse de nuestra conexión a internet a través del servidor de una empresa.

Por último, quisiera sugerir una forma de actuar que considero igualmente segura y sana respecto de las compras en Internet.



La seguridad de nuestras comunicaciones.

Hay miles de lugares en los que podemos comprar artículos desde nuestro PC. Normalmente, sólo necesitaremos una tarjeta de crédito o débito a la que podamos cargar nuestras ciber-compras. Hay muchísimos motivos por los que cualquier persona no querría que su número de tarjeta de crédito circulase ¿libremente? por la red, la seguridad de nuestro dinero es uno de esos motivos.

La sugerencia que propongo es la siguiente: si acostumbramos a realizar ese tipo de compras por Internet, es una buena idea disponer de una cuenta bancaria sólo para esas operaciones. Es decir, además de nuestra cuenta habitual (donde tengamos domiciliados los recibos, las nóminas, etc.) podríamos abrir una cuenta con el saldo mínimo para mantenerla abierta (en muchas entidades ni siquiera es necesario que tenga saldo mínimo). Así, cuando vayamos a hacer una compra por Internet, un método bastante seguro sería realizar una transferencia o traspaso desde nuestra cuenta habitual hasta esta cuenta “especial” únicamente por el valor de aquello que pretendemos comprar. Cuando la transferencia esté realizada, podremos realizar la compra por Internet sin preocupaciones añadidas porque, aunque nuestro número circulase por la red, nadie podría hacer un uso indebido de nuestro dinero porque la cuenta estaría a cero.

NOTA: obviamente, esta forma de proceder funcionará sólo cuando realicemos nuestras compras por internet utilizando una tarjeta de débito, NO UNA DE CRÉDITO, puesto que por su propia naturaleza permitiría la compra incluso sin saldo (como su nombre indica, “a crédito”)

Eso es todo por ahora. Estos consejos, aunque son muy sencillos, pueden ayudar a más de uno a evitarse algún disgusto. Y complicarán la vida hasta extremos insospechados a los amigos de lo ajeno.

Saludetes, linuxeros :)

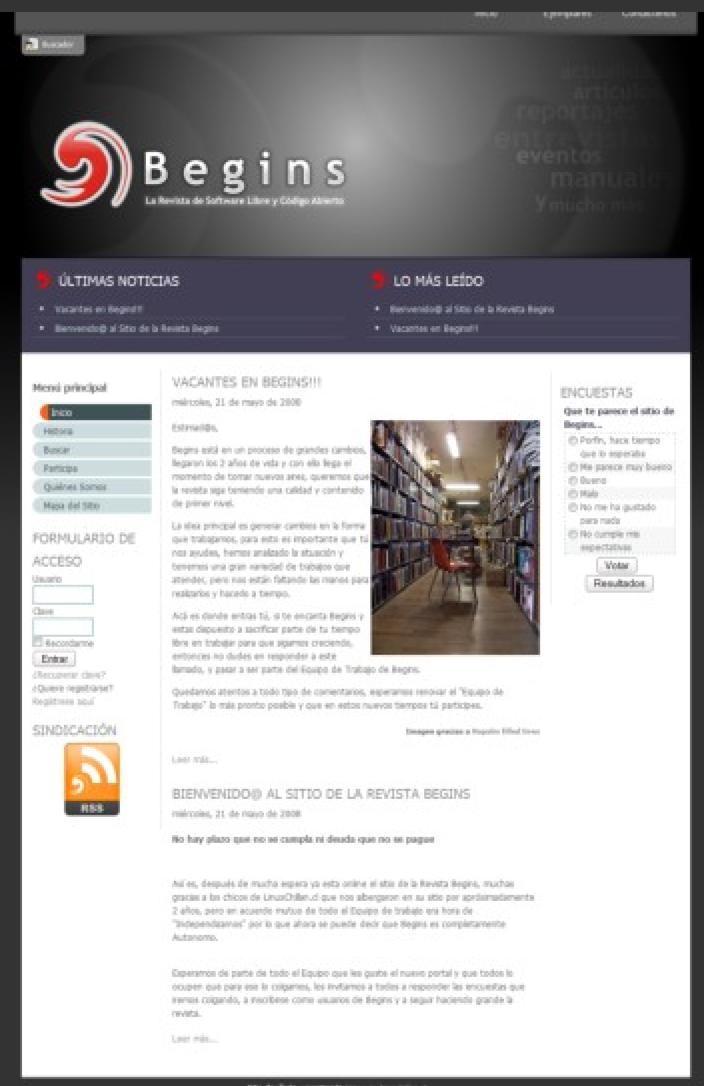
Óscar Calle Gutiérrez
Equipo de Begins



2° Aniversario Revista Begins

Cumplimos un nuevo año de vida con Revista Begins, ya son 2 años de compartir, de conocer a gente extraordinaria, personas dispuestas a colaborar con la difusión del Software Libre en todos sus colores y sabores. A lo largo de estos años se han ido subiendo al carro amigos que ahora son entrañables colaboradores, no solo desde Chile y España, sino también de otras latitudes, personajes que de una u otra forma han estado entregando su servicio, sus comentarios, su buena actitud de trabajo.

Queremos agradecer a todos por compartir este sueño que es Revista Begins, muchas gracias a todos por seguir creyendo en todos nosotros.



Estrenamos Sitio Web

Junto con estos 2 años de vida estrenamos por fin el sitio web de Revista Begins, desde ya los invitamos a visitar y participar del sitio, en donde iremos publicando todas las novedades respecto de cada número, datos freak y varias otras cosas.

Como siempre les decimos "Necesitamos de su ayuda", Begins ha llegado a un punto de crecimiento que a veces deja al equipo de desarrollo un tanto sobrepasado, tenemos mucho trabajo para quienes deseen comprometerse con "la causa" para que sigamos por mucho tiempo más.

Aprovechando la consola para descargar con



Introducción

Bittorrent es un protocolo para el intercambio de archivos, más conocido como Peer to Peer o P2P, es en el último tiempo uno de los protocolos más utilizados para compartir nuestros archivos. Ahora muchas veces disponemos de un servidor remoto con una conexión a Internet de esas que dan miedo y muchas veces no les podemos sacar el mayor provecho ;) a continuación voy a dar una receta para aprovechar el servidor e irnos a la casa y descargar nuestros archivos mientras dormimos sin problema.

Ingredientes :

- **Servidor Linux**
- **Acceso SSH a ese servidor**
- **Screen**
- **Bittorrent**

Screen es el programa que hace la magia en nuestra terminal, lo que nos permite es generar terminales virtuales por lo que podemos dejar descargando nuestro bittorrent y que funcione todo el tiempo que nosotros queramos y podemos recuperar la terminal con un simple comando.

Primero que todo, ingresamos a nuestro servidor :

```
# ssh <nombre_de_usuario>@127.0.0.1
```

Nos pedirá la contraseña del usuario que ingresamos antes del @ (recordar que <nombre_de_usuario> se remplaza completamente por el nombre de usuario en tu servidor.)



Luego, creamos una terminal virtual con screen

screen -t bit

con ese comando creamos la terminal virtual e inmediatamente notamos que algo ha sucedido por el pestañeo que sucede, luego es hora de poner en marcha el bittorrent y dejar funcionando nuestra descarga.

btlaunchmanycurses <nombre_archivo.torrent> --minport 8000 --maxport 8003 --max_upload_rate 16

Con este comando lanzamos el bittorrent, le indicamos los puertos a utilizar, sin añadirlos el bittorrent no me funciona además le indicamos un máximo de upload para que ande bien la descarga.

Una vez que ya tenemos funcionando el bittorrent, en nuestra pantalla vemos como esta funcionando y descargando nuestros archivos, ahora para salir y que todo siga funcionando en screen tenemos que hacerlo con el siguiente comando : ***CTRL + a + d***

```
root@apu: ~
Archivo  Editar  Ver  Terminal  Solapas  Ayuda
 Filename                                  Size    Download    Upload
 21[2008]DvDrip-aXXo                    701.7 M    9.9 K/s  15.8 K/s
 ^---  finishing in 5:24:35 (57.3%)             402.0 M    403.9 M

                                      Totals:    9.9 K/s  15.8 K/s
                                                402.0 M    403.9 M
 New torrent: 21[2008]DvDrip-aXXo.4242875.TPB.torrent
```

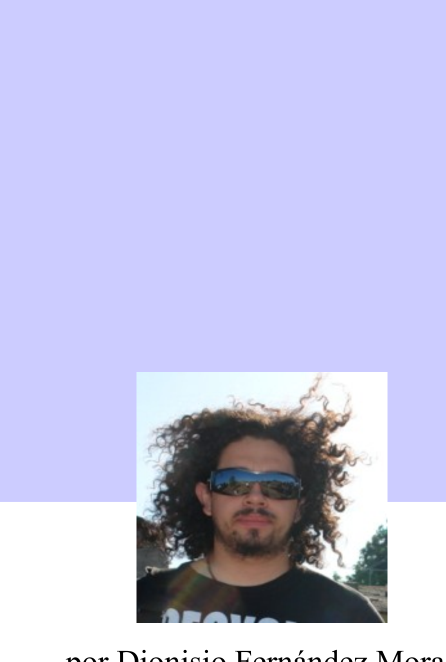


Una vez hayamos vuelto a nuestra terminal original, en backround nuestra descarga seguirá funcionando. Ahora si queremos volver a nuestra terminal virtual para saber como va nuestra descarga solo tenemos que aplicar el siguiente comando :

```
#screen -t bit -D -r
```

Recordar que cuando se salga de una terminal virtual hay que hacerlo con *CTRL+a+b*

Con esta simple receta podemos sacar provecho a las bondades de las aplicaciones que nos entrega el Software Libre.



por Dionisio Fernández Mora
<zeroblack@linuxchillan.cl>

Entendiendo Web Services con MonoDevelop I de IV

Introducción

En las organizaciones un tema importante hoy día, es la integración de los sistemas de información ya existentes, para tener una mejor comunicación entre ellos y hacia otros sistemas externos en otras organizaciones y por supuesto hacia la Internet. A través de los años entre las distintas soluciones para la comunicación distribuida se crearon tecnologías de componentes como Microsoft DCOM y Corba que ayudaban a este propósito, Sin embargo estas tecnologías tienen ciertas limitantes como el hecho de ser dependientes del sistema operativo, la falta de compatibilidad entre protocolos y la dificultad para la implementación y el mantenimiento, etc. Estas limitantes hacen que no sean una opción adecuada para la comunicación y la integración de sistemas en distintas plataformas o sistemas localizados en la Internet.
En este escenario los Web Services (servicios Web) son la solución adecuada ya que permiten la integración de sistemas de una forma estándar, práctica y transparente.

¿Que son los Web Services?

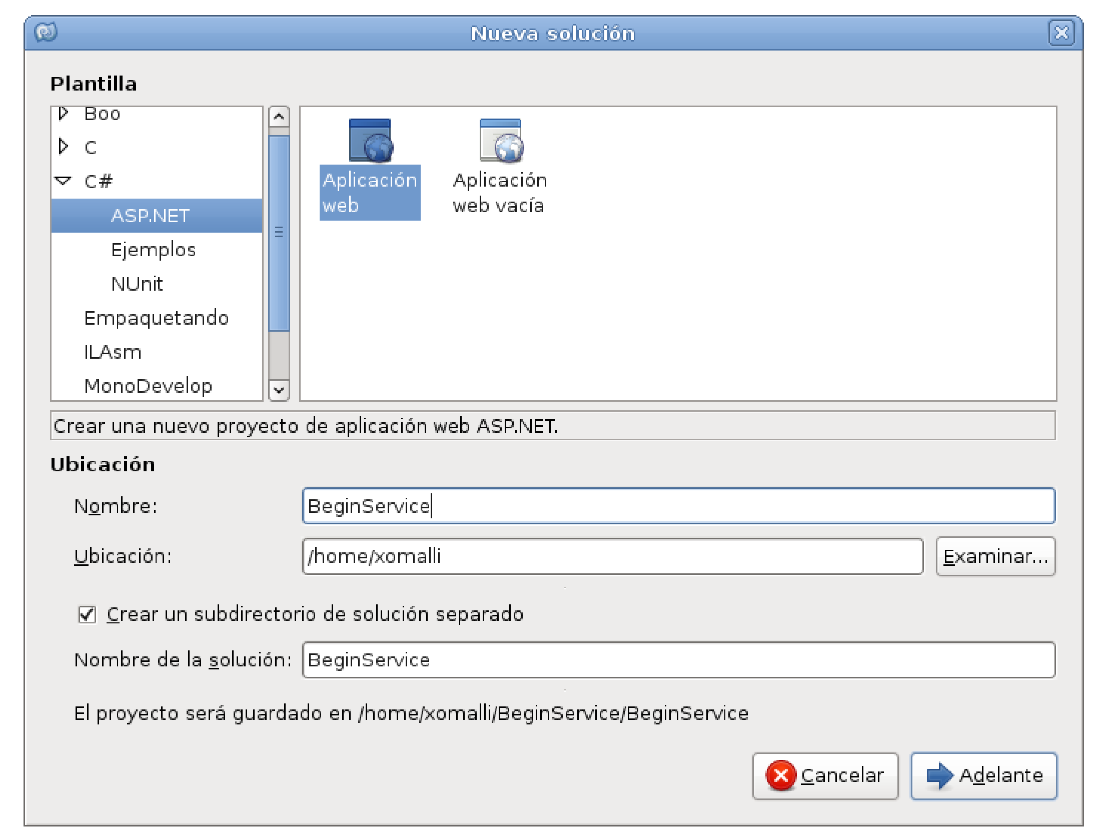
Un Web Service es un componente de software que se publica en un servidor Web o HTTP y que permite la comunicación distribuida entre sistemas o aplicaciones en diferentes máquinas al exponer uno o varios métodos que al ser invocados por otros sistemas vía un URL devuelven información usando protocolos estándar como HTTP y XML. Este documento XML de información devuelto por el Web Service puede ser procesado por cualquier aplicación o sistema en cualquier sistema operativo con cualquier lenguaje de programación que tenga las capacidades de procesar XML.

Antes de empezar debemos tener instalado y configurado el entorno de desarrollo (IDE) Monodevelop http://www.monodevelop.org/Main_Page .que contiene todas las herramientas necesarias para construir y consumir servicios Web, a nivel de programación un Web Service es una clase que generalmente hereda de System.Web.Services.WebService y que contiene uno o varios métodos marcados como `[WebMethod()]` que serán publicados en una página con extensión .*asmx* que contiene directivas de ASP.NET y que es la extensión predeterminada de las páginas donde se ejecutan los Web Service.

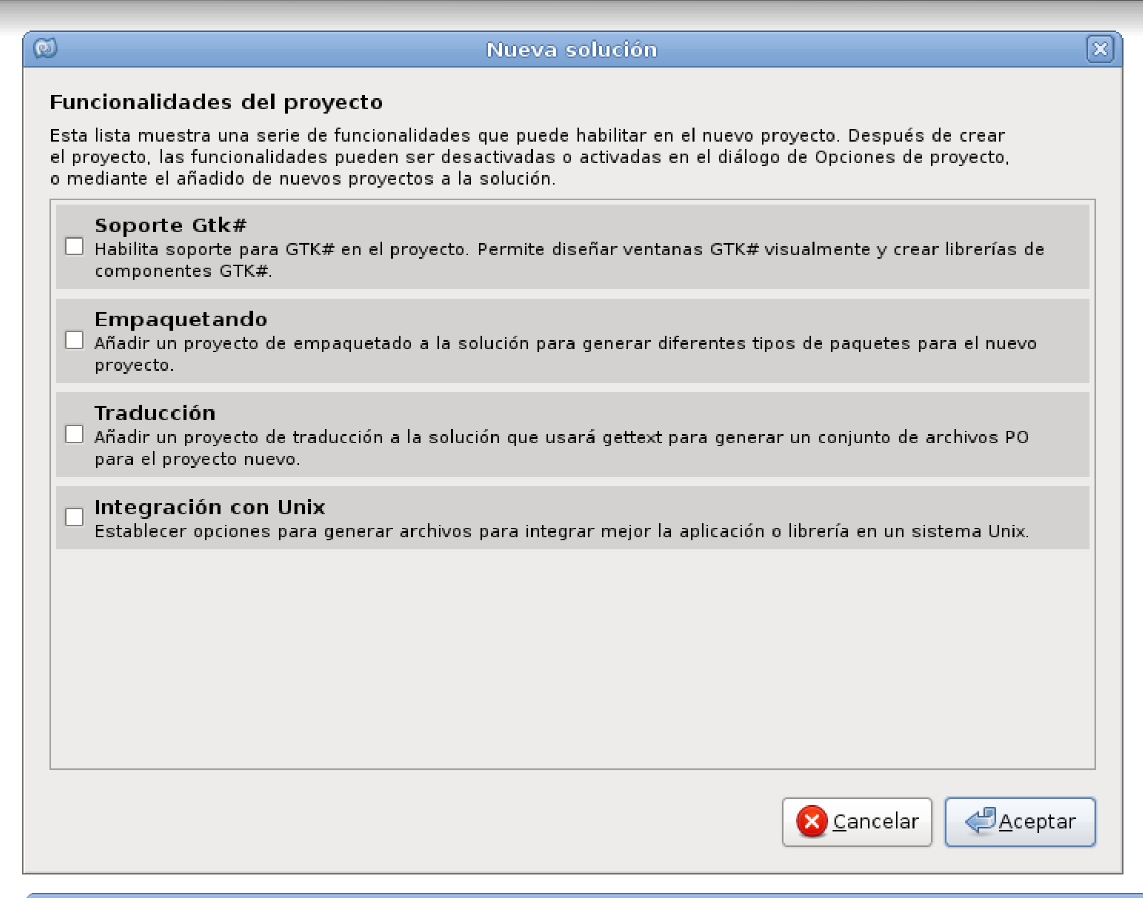
Una diferencia entre las páginas ASPX y los Web Services es que los Web Services no están pensados para ser utilizados por usuarios finales sino para ser usados por aplicaciones, por lo que no requieren una interfaz gráfica, aunque Mono provee una plantilla que se utiliza en cada Web Service para probar su ejecución de manera local.

Empezaremos dentro de la ventana principal de MonoDevelop, .ahí seleccionaremos en el menú *Archivo* el submenú de Nueva solución.

MonoDevelop
Archivo Editar Ver Buscar Proyecto Herramientas Ventana Ayuda
Nuevo archivo... Ctrl+N
Nueva solución... Ctrl+Shift+N
Abrir... Ctrl+O
Obtener...
Abrir archivo de la solución... Alt+Shift+O
Guardar Ctrl+S
Guardar como... Ctrl+Shift+S
Guardar todo
Vista previa de la impresión Ctrl+Shift+P
Imprimir... Ctrl+P
Cerrar archivo Ctrl+W
Cerrar todos Ctrl+Shift+W
Cerrar solución
Recargar archivo Ctrl+U
Archivos recientes
Soluciones recientes
Salir Ctrl+Q
Ar... Sol... Cla...
ln 13 col 4 ch 1 INS



Al hacer clic se aparecerá la siguiente ventana, en la lista de opciones del lado izquierdo abajo de la etiqueta *Plantilla,* seleccionamos *C#* y luego seleccionamos *ASP.NET* y en las opciones del lado derecho seleccionamos el icono *Aplicación Web ,* Tecleamos un nombre. para el proyecto así como una ubicación para que Monodevelop haga la estructura, pulsamos el botón *Adelante* para continuar con la siguiente ventana.

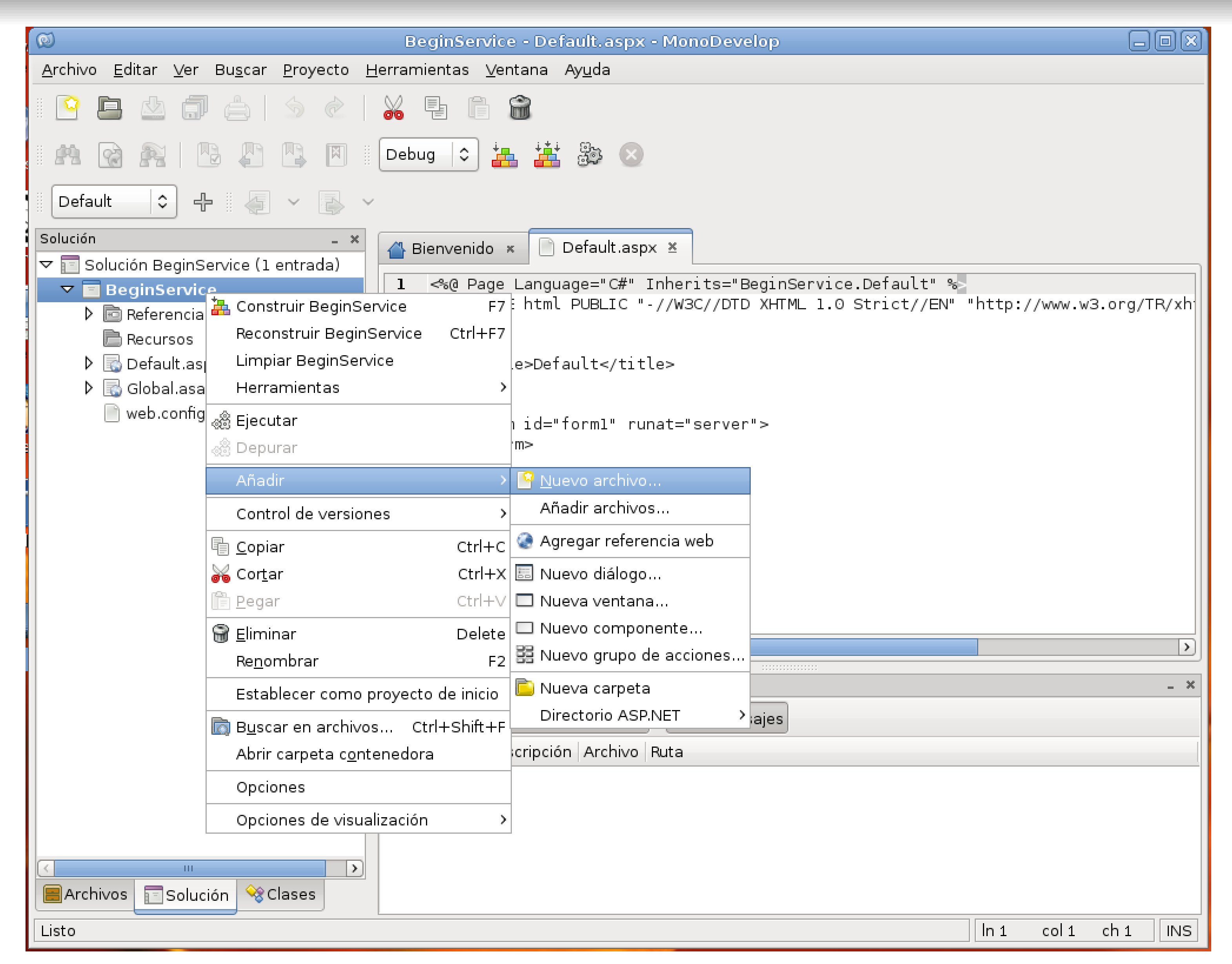


Al hacer clic se aparecerá la siguiente ventana, en la lista de opciones del lado izquierdo abajo de la etiqueta *Plantilla,* seleccionamos *C#* y luego seleccionamos *ASP.NET* y en las opciones del lado derecho seleccionamos el icono *Aplicación Web* , Tecleamos un nombre. para el proyecto así como una ubicación para que Monodevelop haga la estructura, pulsamos el botón *Adelante* para continuar con la siguiente ventana.

BeginService - Default.aspx - MonoDevelop

```
<%@ Page Language="C#" Inherits="BeginService.Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xh
<html>
<head>
    <title>Default</title>
</head>
<body>
    <form id="form1" runat="server">
    </form>
</body>
</html>
```

Monodevelop nos creará la estructura de nuestra aplicación Web, como se ve en la pestaña solución del panel izquierdo.

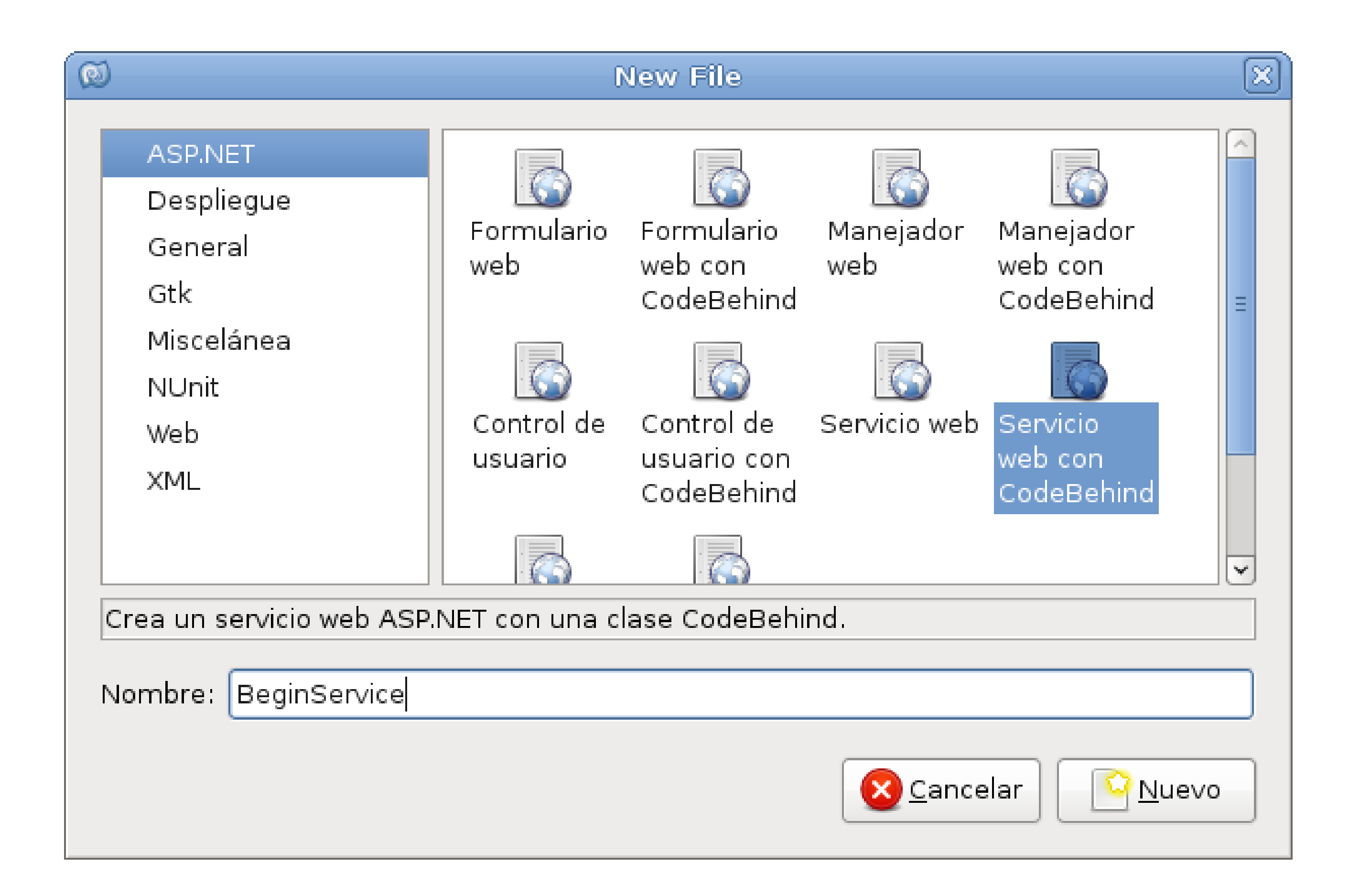


Bien ahora agregaremos a nuestra solución un archivo Web Services, esto lo hacemos haciendo clic con el botón derecho del ratón, seleccionando el nombre en negritas del proyecto, como se muestra en la imagen.

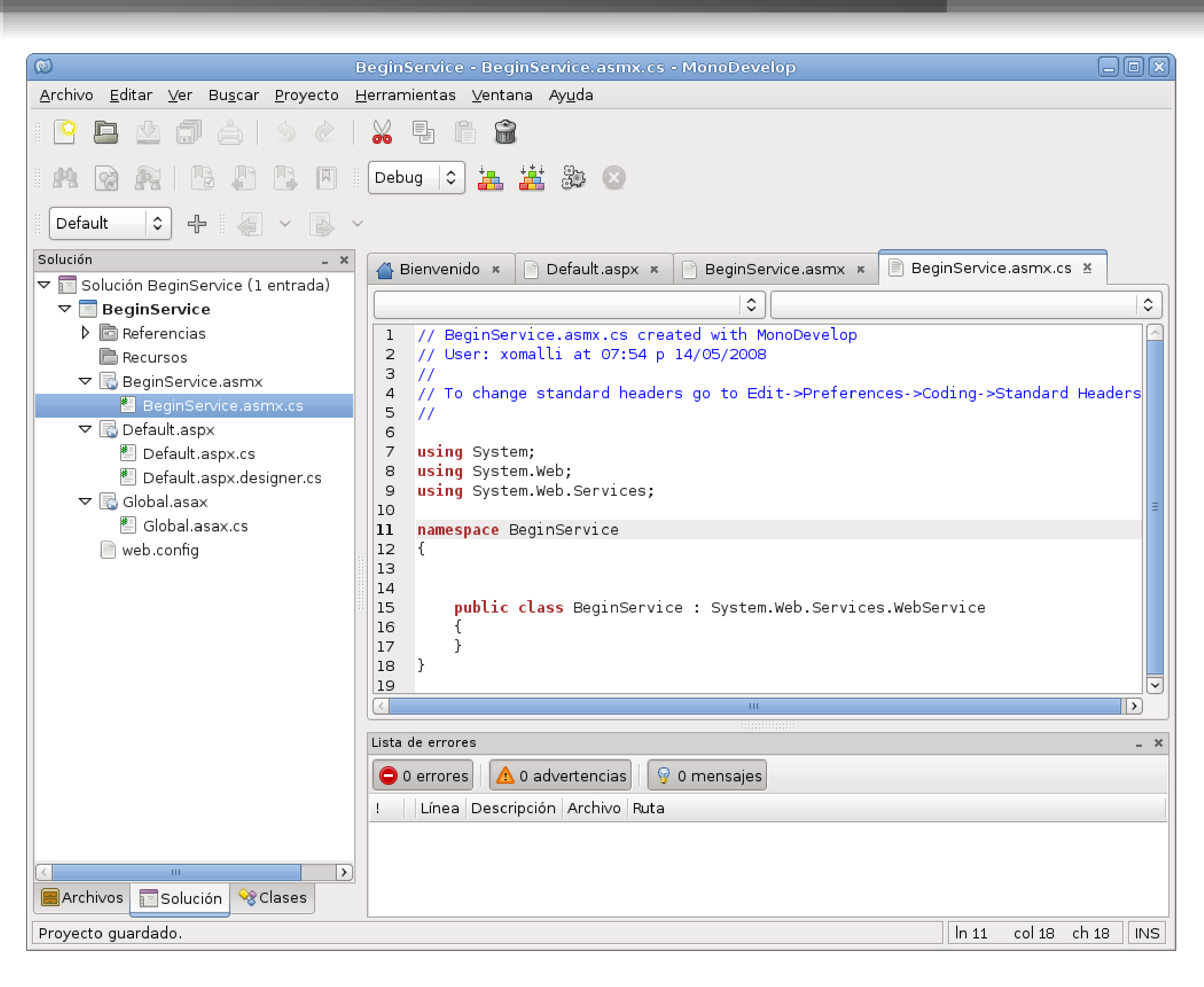
Code Behind y Scripting Language

En la ventana *New File* seleccionaremos en la lista del panel izquierdo la opción *ASP.NET* y en el panel de iconos del lado derecho la opción *Servicio web con CodeBehind* y en el cuadro de texto de la parte inferior tecleamos el nombre del *Web Service* en este caso *BeginService*.

Las páginas ASP .NET *(aspx)* y los Páginas de los servicios Web *(asmx)* pueden programarse como *Server Side Scripting* que es una mezcla de etiquetas *aspx* con código C# dentro de una misma página con extensión *.asmx*, la otra forma *Code Behind* es la más recomendable ya que permite separar el funcionamiento o sea la lógica de programación en un archivo de clase C# y la presentación en la página *.asmx*, a la cual se asociada el archivo de clase C#.



En este punto **Monodelop** nos mostrará la siguiente estructura de archivos de nuestra aplicación Web incluido el archivo de publicación *BeginService.asmx* para nuestro servicio Web y el archivo de clase *BeginService.asmx.cs* asociado a ese archivo.



En el archivo de clase **Monodevelop** nos crea la estructura mínima necesaria que debe tener cualquier clase que sirva como Web Services, primeramente la clase deberá de heredar de la clase WebService que se encuentra en el ensamblado `System.Web.Services`

```
public class BeginService : System.Web.Services.WebService
```

Ahora completaremos esta clase con métodos, cada método que sera consumido al publicarse el Web Services, deberá de llevar el atributo `[WebMethod()]` arriba de su declaración, los parámetros y los valores devueltos por los métodos en los Web Services pueden ser desde tipos de datos primitivos como `long`, `bool`, `float` hasta objetos propios del Framework como un `DataSet`.

Listado 1.1 El código final de la clase BeginService

```
using System;
using System.Web;
using System.Web.Services;
using System.Diagnostics;

namespace BeginService{
	[WebService
	(	Name = "BeginService",
		Description = "BeginService",
		Namespace = "http://www.BeginService.example"
	)
	]
	public class BeginService : System.Web.Services.WebService{
		public BeginService(){}
		[WebMethod(Description="Regresa el máximo común divisor de 2 números")]
		public long MaxComunDivisor(long m,long n){
			long r;
			while(n != 0){
				r = m % n;
				m = n;
				n = r;
			}
			return m;
		}
		[WebMethod(Description="Regresa una cadena para representar una fecha
larga")]
		public string getFecha(){
			DateTime dthoy = DateTime.Today;
			string[] meses =
{"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio","Agosto","Septiembre","Octubre"
,"Noviembre","Diciembre"};
			return dthoy.Day.ToString() + " de " + meses[dthoy.Month -1] + " del " +
dthoy.Year;
		}
		[WebMethod(Description="Regresa true si la cadena es número, false si no lo
es.")]
		public bool isAnumber(string s){
			try{
			return Single.IsNaN(float.Parse(s)) ? false : true;
			}catch(FormatException e){
				Debug.WriteLine(e.Message);
				return false; }
		}
	}
}
```

Este ejemplo muestra el atributo **WebMethod** con la propiedad Description la cual nos sirve para mostrar la descripción del método cuando el Web Services sea publicado.

```
[WebMethod(Description="Regresa el máximo común divisor de 2 números")]
public long MaxComunDivisor(long m,long n)
```

Otras propiedades del atributo **WebMethod** son:

BufferResponse	true o false para indicar o no si la respuesta del Web Services es serializado en un buffer de memoria hasta ser completada o hasta que el buffer este lleno.
CacheDuration	Establece el número de segundos que la respuesta estará contenida en el cache.
EnableSession	false o true para el soporte de sesiones.
MessageName	Identifica mediante un alias los métodos sobrecargados.
TransactionOption	Habilita al método para participar o no en una transacción, si ocurre una excepción en el método la transacción es abortada.

Antes de compilar el proyecto nos aseguramos de agregar al archivo *Default.aspx* un enlace a nuestro Web Service entre las etiquetas *<form></form>* de la siguiente manera:

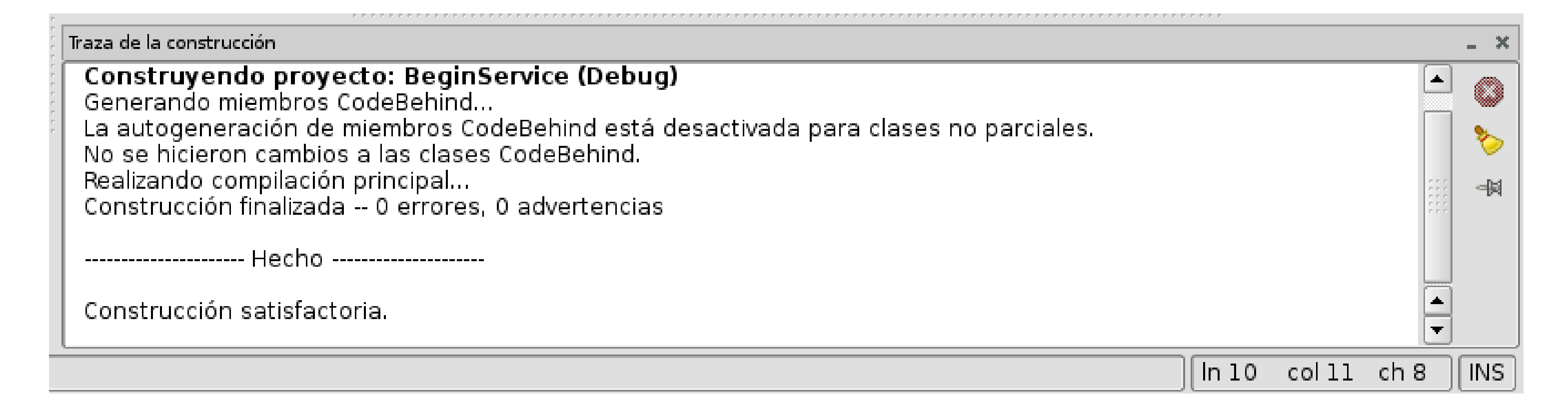
```
<form id="form1" runat="server">
      <a href="BeginService.asmx">Begin Service</a>
      </form>
```

Finalmente el archivo *BeginService.asmx* quedará tal y como **Monodelop** lo genero.

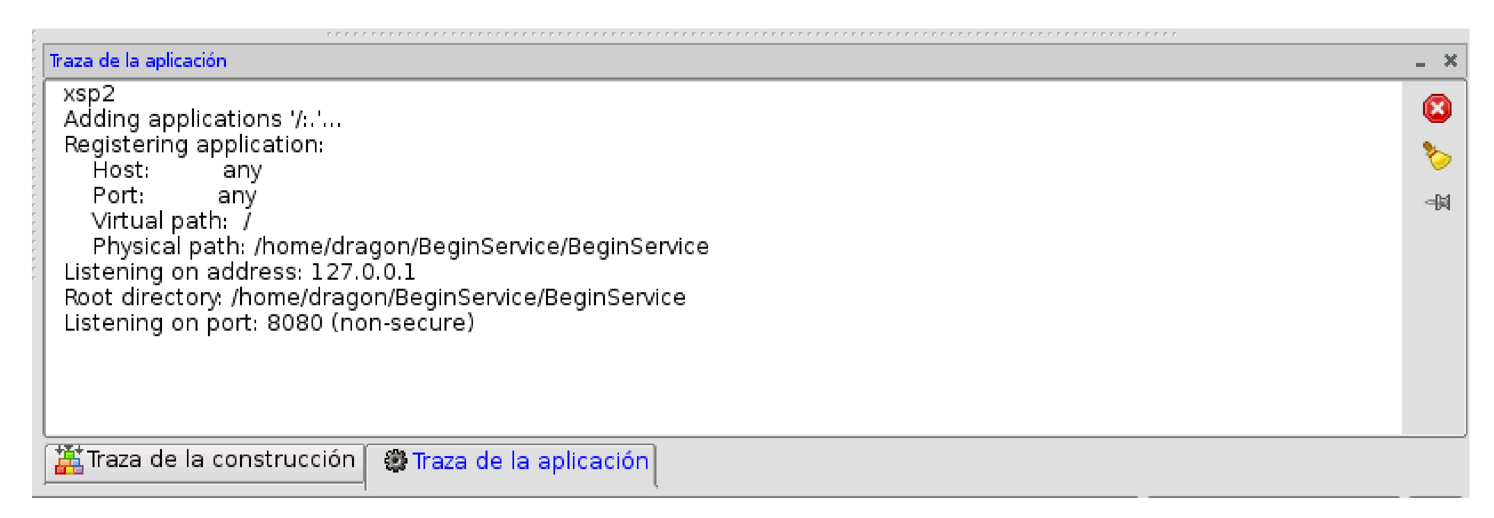
```
<%@ WebService Language="C#" Class="BeginService.BeginService" %>
```

Compilamos el proyecto con la opción del submenú *Construir BeginService* debajo del menú *Proyecto* o pulsando la tecla F7.

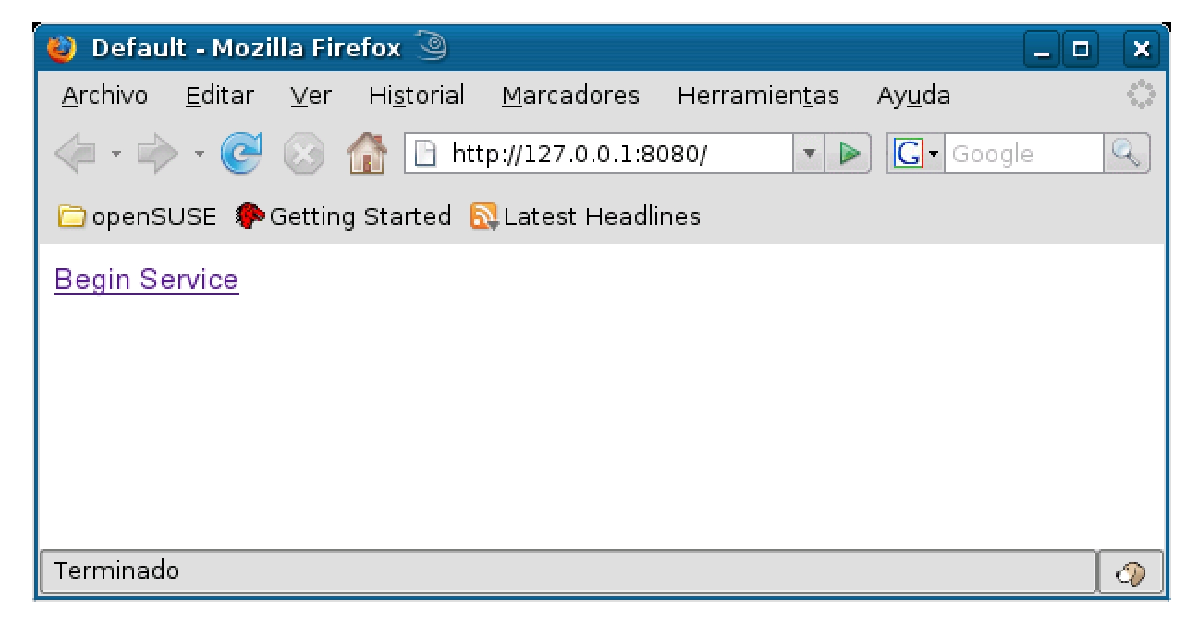
Si todo se ejecuta sin errores **Monodevelop** mostrará en la parte de abajo el siguiente mensaje:



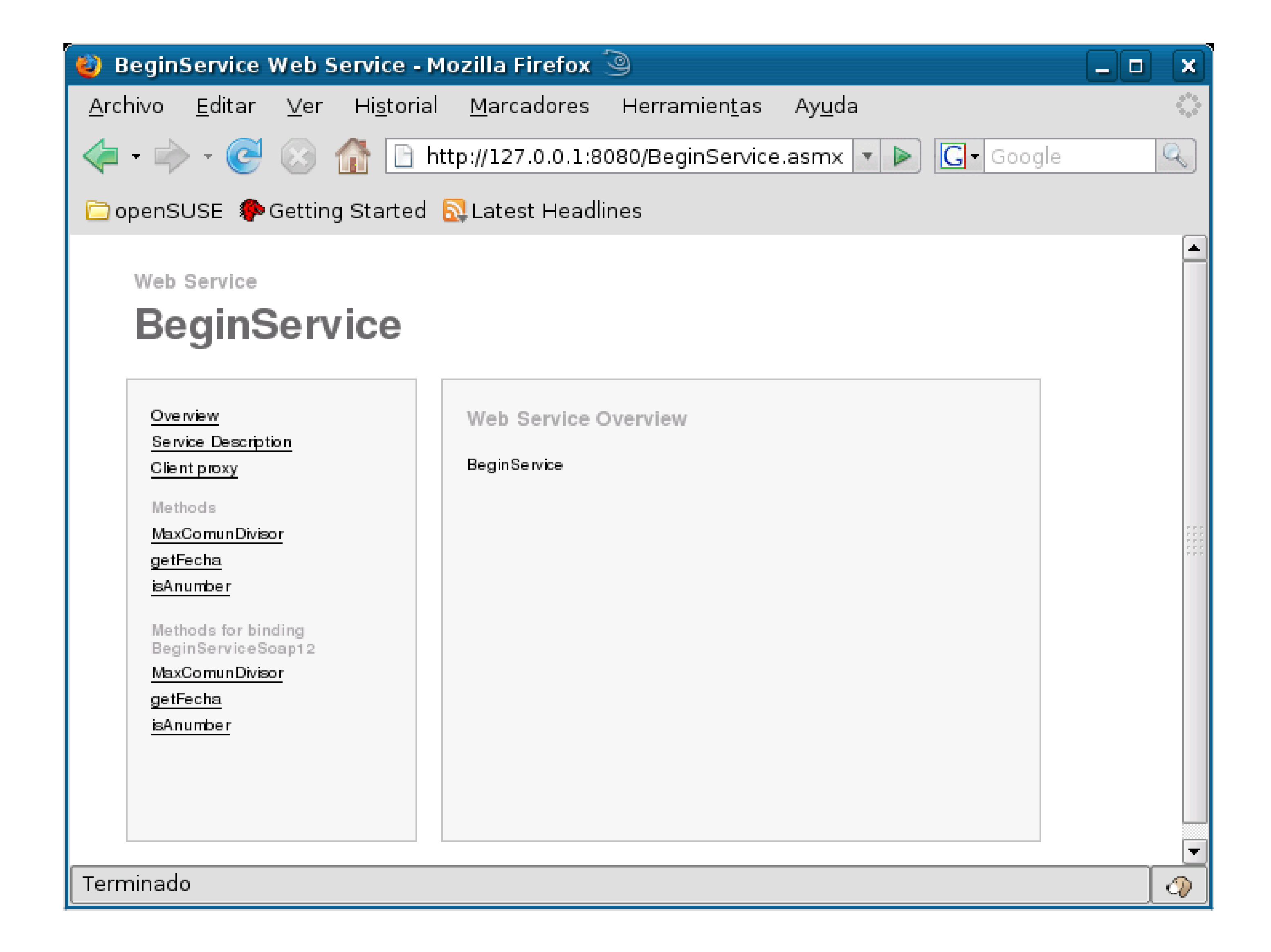
Ahora ejecutaremos el proyecto con la opción del submenú *Ejecutar* del menú *Proyecto* o pulsando la tecla F5.
Al ejecutar el proyecto se ejecutará el comando *xsp2* que es el servidor Web para desarrollo de aplicaciones en el Framework **Mono.**



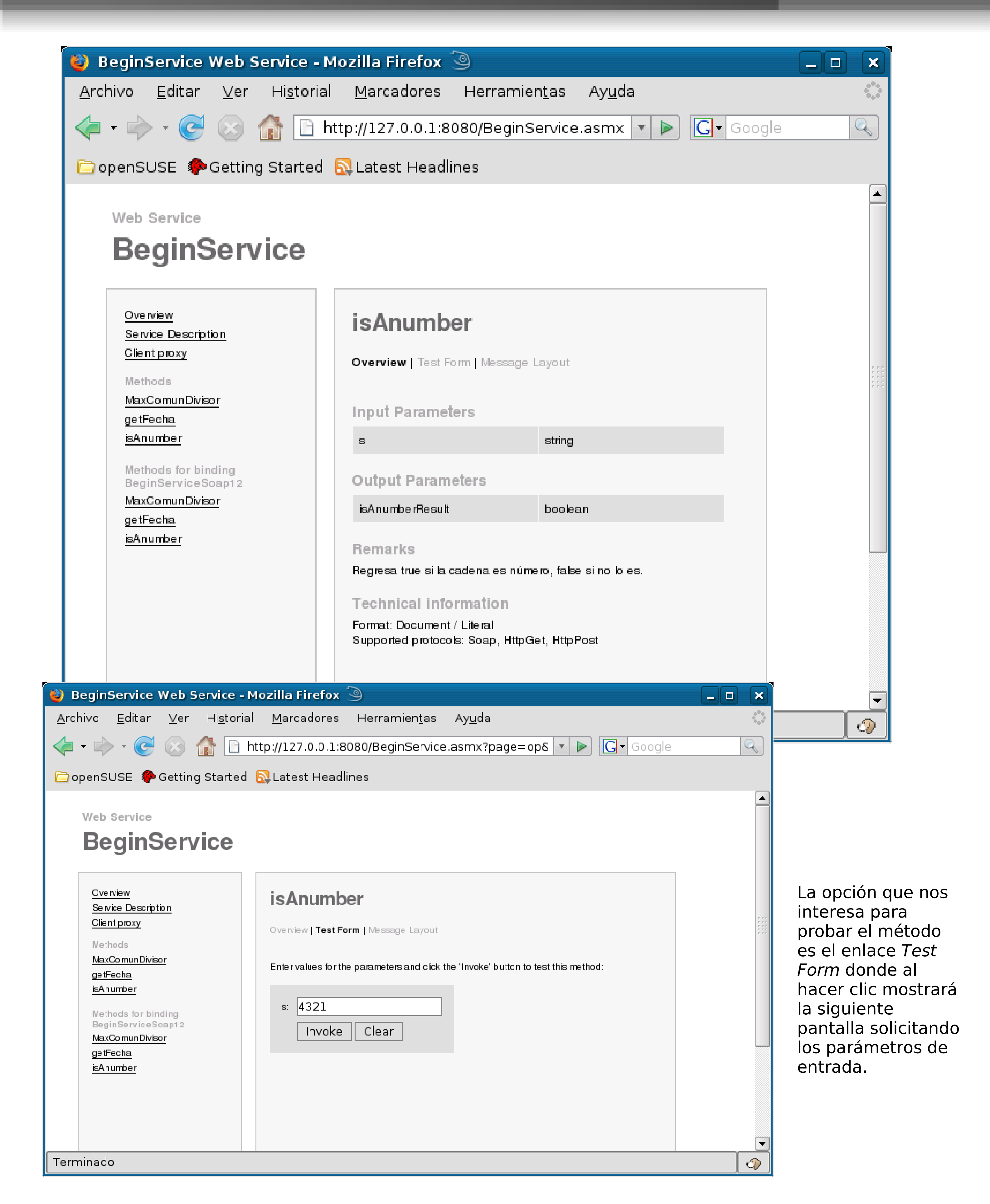
Después **Monodevelop** nos abrirá una ventana de firefox (siempre que sea el navegador predeterminado)con el siguiente URL para mostrarnos el resultado final.



Si todo se ejecutó correctamente al hacer clic en el enlace se mostrará la página predeterminada de **Mono** para probar los servicios Web de manera local, en esta página veremos en la tabla del lado izquierdo veremos sus propiedades como enlaces, entre ellas cada método que definimos en el archivo de clase.



Para probar un método en particular hacemos clic en el enlace correspondiente, ahí nos mostrará las opciones de ese método como enlaces en la parte superior y como *overview* nos muestra sus parámetros y el tipo de dato que devuelve..



La opción que nos interesa para probar el método es el enlace *Test Form* donde al hacer clic mostrará la siguiente pantalla solicitando los parámetros de entrada.

Finalmente al presionar el botón *invoke* nos mostrará el mensaje SOAP que será procesado por cualquier cliente con capacidades para interpretar XML independientemente del lenguaje de programación o del sistema operativo.

BeginService Web Service - Mozilla Firefox

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

http://127.0.0.1:8080/BeginService.asmx?page=op&tab

openSUSE Getting Started Latest Headlines

Web Service

BeginService

Overview
Service Description
Client proxy

Methods
MaxComunDivisor
getFecha
isAnumber

Methods for binding BeginServiceSoap12
MaxComunDivisor
getFecha
isAnumber

isAnumber

Overview | **Test Form** | Message Layout

Enter values for the parameters and click the 'Invoke' button to test this method:

s:

Invoke Clear

The web service returned the following result:

```
<?xml version="1.0" encoding="utf-8"?>
<boolean xmlns="http://www.BeginService.example">true</boolean>
```

Terminado

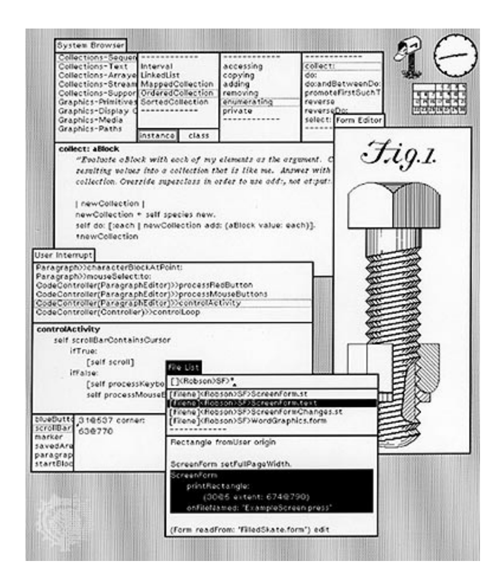
CONCLUSIÓN

En esta primera parte vemos como **Monodevelop** y el Framework **Mono** nos proporcionan toda la infraestructura necesaria para crear Web Services, en la segunda parte mostraré como diferentes clientes pueden consumir o utilizar Web Services con relativa facilidad, como referencia consultar http://www.mono-project.com/Web_Services y la documentación del proyecto Mono en http://www.go-mono.com.



Los ejemplos pueden ser descargados de http://www.humansharp.com/index.php?var=code

por Martín Márquez xomalli@gmail.com



LOS AMBIENTES DE ESCRITORIO EN LINUX

El comienzo

Quiero comenzar este artículo haciendo un reconocimiento a quienes iniciaron una nueva era en la informática, con la invención de las interfaces gráficas.

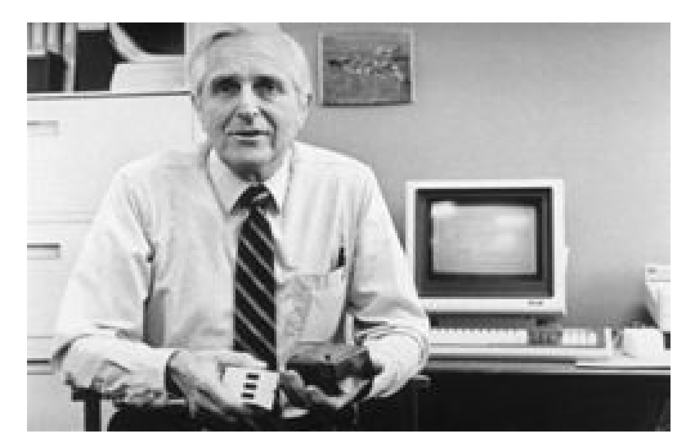
Xerox PARC (*Palo Alto Research Center*, ‘Centro de Investigación de Palo Alto’) era una división de investigación de Xerox Corporation, con sede en Palo Alto, California, EE.UU. Fue fundado en 1970 como consecuencia directa del fenomenal éxito de la empresa y ampliado como compañía independiente (aunque propiedad de Xerox) en 2002.

Es famoso por crear esencialmente el paradigma moderno de interfaz gráfica de usuario (GUI) para los computadores personales.

En el PARC nacieron los iconos, las representaciones gráficas de las órdenes que le podemos dar al computador, las ventanas, las áreas de trabajo que permiten delimitar zonas en la pantalla, y los gráficos gracias a los mapas de bits, una tecnología que permitía visualizar gráficos en la pantalla de nuestros computadores con gran precisión.

El entorno se denominó **PARC User Interface** y en él se definieron los conceptos de ventanas, *checkbox*, botones de radio, menús y puntero del *mouse*. Fue implementado comercialmente en el Xerox Star 80107.

Pero en el PARC también se utilizarían por primera vez, además del teclado y el *mouse* como principal mecanismo de control del computador, un invento de Douglas Engelbart; nacería la red Ethernet, la posibilidad de conectar entre sí los computadores personales; surgirían las primeras impresoras láser que permitirían posteriormente el nacimiento de la autoedición; y también verían la luz la programación orientada a objetos mediante el lenguaje Smalltalk, pero eso es otra historia.



Engelbart, en su oficina de Tymshare, muestra el mouse original junto al entonces más moderno modelo de 1984.

X Window System

El Sistema de Ventana X fue desarrollado a mediados de los años 1980 en el MIT para dotar de una interfaz gráfica a los sistemas Unix. Este protocolo permite la interacción gráfica en red entre un usuario y una o más computadores haciendo transparente la red para éste. Generalmente se refiere a la versión 11 de este protocolo, X11, el que está en uso actualmente.

X es el encargado de mostrar la información gráfica y es totalmente independiente del sistema operativo. El sistema de ventana X distribuye el procesamiento de aplicaciones especificando enlaces cliente-servidor. El servidor provee servicios para acceder a la pantalla, teclado y ratón, mientras que los clientes son las aplicaciones que utilizan estos recursos para interacción con el usuario. De este modo mientras el servidor se ejecuta de manera local, las aplicaciones pueden ejecutarse remotamente desde otras máquinas, proporcionando así el concepto de transparencia de red.

Debido a este esquema cliente-servidor, se puede decir que X se comporta como una terminal gráfica virtual.

El hecho que exista un estándar definido para X permite que se desarrollen servidores X (XServers) para distintos sistemas operativos, plataformas, Hardwares, etc... lo que hace que el código sea muy portable. Por ejemplo. permite tener Xclients ejecutándose en un potente servidor UNIX mientras los resultados son visualizados en un PC de escritorio con cualquier otro sistema operativo funcionando.

La comunicación entre el Xclient y el Xserver se realiza por medio de un protocolo conocido como Xprotocol, que constituye una serie de bytes interpretados como comandos básicos para generar ventanas, posicionarlas, o controlar eventos.

Sin embargo los aspectos de decoración de ventana y manejos de ventanas no están definidos en Xlib. X no es un gestor de ventanas, necesita de uno para controlar el manejo de ventanas. Esto trae la ventaja que permite al usuario instalar el administrador de ventanas que más le agrade, e incluso tener varios instalados eligiendo el más apropiado a la hora de acceder a X. También trae la ventaja que hace de X estrictamente un sistema gráfico, de tal modo que un Xclient podría estar enviando un gráfico a una pantalla, a una impresora o a cualquier otro hardware sin darse cuenta, flexibilizando la salida gráfica.



Logo de X Window System.

Entorno de escritorio

Un entorno de escritorio (en inglés, *Desktop Environment*) es un conjunto de software para ofrecer al usuario de un computador una interacción amigable y cómoda.

El software es una solución completa de interfaz gráfica de usuario o GUI (*Graphical User Interface*), ofrece iconos, barras de herramientas, botones, ventanas, barras, programas e integración entre aplicaciones con habilidades como, arrastrar-y-soltar (*drag&drop*).

En general cada entorno de escritorio se distingue por su aspecto y comportamiento particulares, aunque algunos tienden a imitar características de escritorios ya existentes.

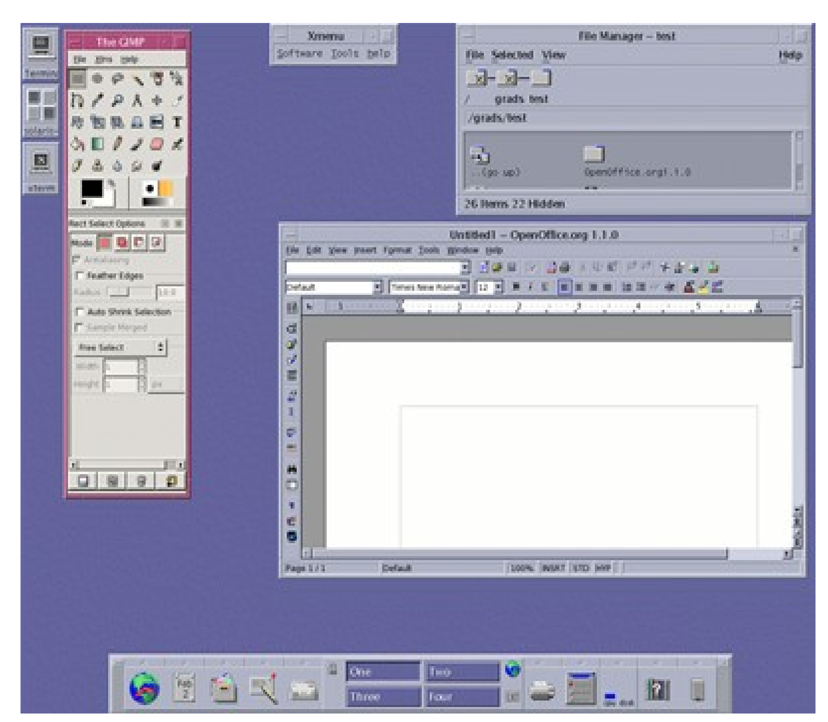
El primer entorno moderno de escritorio que se comercializó fue desarrollado por Xerox en los años 1980.

Actualmente el entorno más conocido es el ofrecido por la familia Windows aunque existen otros como los de Macintosh (Classic y Cocoa) y de código abierto (o software libre) como **GNOME**, **KDE** o **Xfce**.

Common Desktop Environment

CDE (*Common Desktop Environment*) es un entorno de escritorio gráfico para Unix, basado en la biblioteca Motif. CDE fue desarrollado en conjunto por Hewlett-Packard, IBM, Novell y Sun Microsystems.

Hasta aproximadamente el año 2000, CDE era considerado el escritorio estándar para los sistemas Unix, pero al mismo tiempo, escritorios de software libre como KDE y GNOME fueron ganando madurez y convirtiéndose en partes esenciales de la plataforma GNU/ Linux. Hacia el 2001, Hewlett-Packard y Sun Microsystems anunciaron que reemplazarían CDE por GNOME para sus sistemas Unix.



CDE fue desarrollado en conjunto por Hewlett-Packard, IBM, Novell y Sun Microsystems.

LOS ESCRITORIOS EN LINUX



KDE

K Desktop Environment o Entorno de Escritorio K, es un entorno de escritorio e infraestructura de desarrollo para sistemas Unix/Linux.

De acuerdo con su página web, «KDE es un entorno de Escritorio contemporáneo para estaciones de trabajo Unix. KDE llena la necesidad de un escritorio amigable para estaciones de trabajo Unix, similar a los escritorios de MacOS o Windows».

La «K», originariamente, representaba la palabra «Kool», pero su significado fue abandonado más tarde. Actualmente significa simplemente «K».

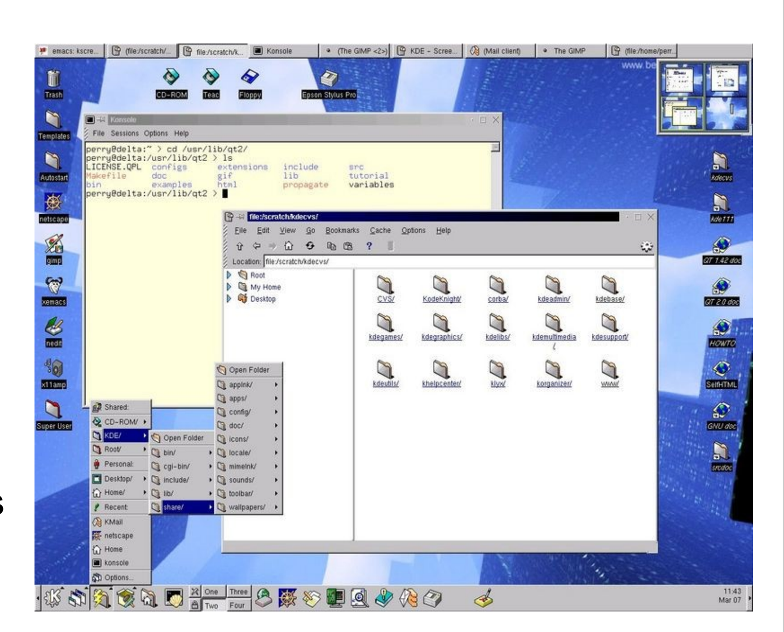
La mascota del proyecto es un pequeño dragón llamado Konqi.

Historia

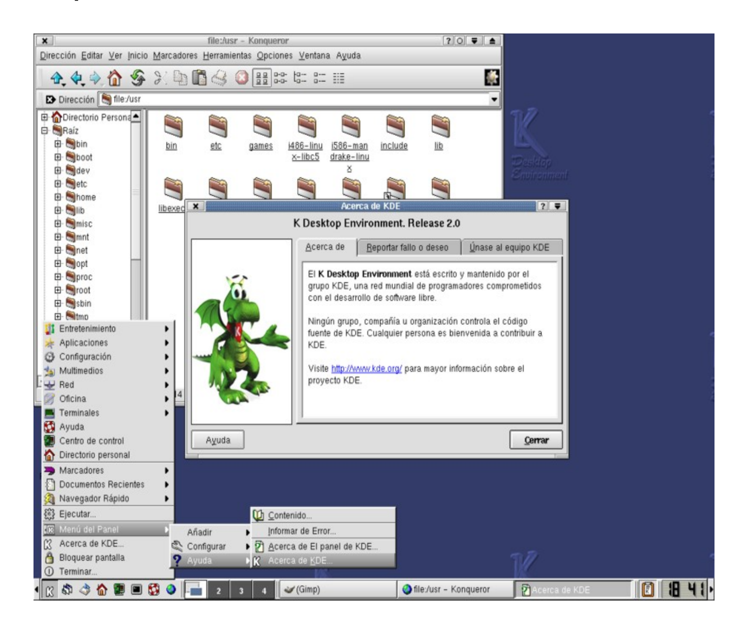
El proyecto fue iniciado en octubre de 1996 por el programador alemán Matthias Ettrich, quien buscaba crear una interfaz gráfica unificada para sistemas Unix. En sus inicios imitó a CDE, un entorno utilizado por varios Unix.

Dos factores llevaron a la creación del proyecto alternativo GNOME en 1997: la elección de la biblioteca Qt, que por aquel entonces poseía una licencia incompatible con la GPL de GNU, aunque libre: la QPL, y en menor medida la importancia del lenguaje C++ para el desarrollo de KDE. La rivalidad actual entre ambos proyectos se considera beneficiosa generalmente y existe, de hecho, una constante cooperación e inspiración mutua.

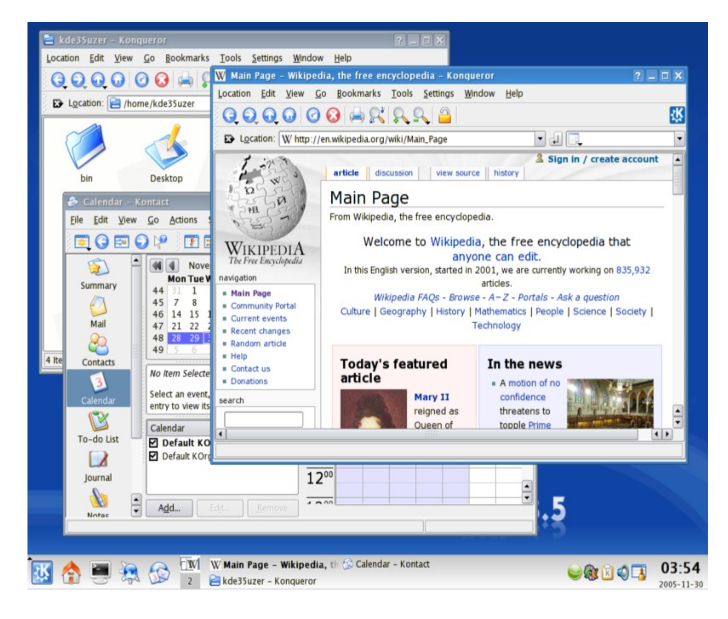
KDE 1: Al año siguiente, se publicó KDE 1.0. Esta versión contenía un panel (barra de tareas y lanzador de aplicaciones), un escritorio sobre el cual dejar iconos, un administrador de archivos (Kfm) y un gran número de utilidades.



KDE 2: Lanzado en el año 2000, fue reescrito casi por completo. Esta versión incluía Konqueror (un navegador web y gestor de archivos) además de muchas nuevas tecnologías con el objetivo de mejorar la integración entre aplicaciones. En esta versión mejoró parcialmente el aspecto visual.



KDE 3: Fue publicado en el año 2002, y es la evolución de KDE 2. El aspecto de la interfaz no varió hasta KDE 3.1, en el que consta una importante mejora referente al tema visual: Keramik es incluido como nuevo tema junto con el conjunto de iconos Crystal GT.



KDE 4: Está basado en la cuarta versión de Qt, que en principio incrementa el rendimiento respecto a la versión anterior. La renovación de las librerías y las herramientas de compilación también facilitarán el soporte para plataformas no basadas en X11, incluyendo Microsoft Windows y Mac OS X, ya que uno de los objetivos de KDE 4 es que pueda ser portado más fácilmente a diferentes sistemas operativos.





GNOME

GNOME es un entorno de escritorio para sistemas operativos de tipo Unix bajo tecnología X Window. Forma parte oficial del proyecto GNU. Nació como una alternativa a KDE.

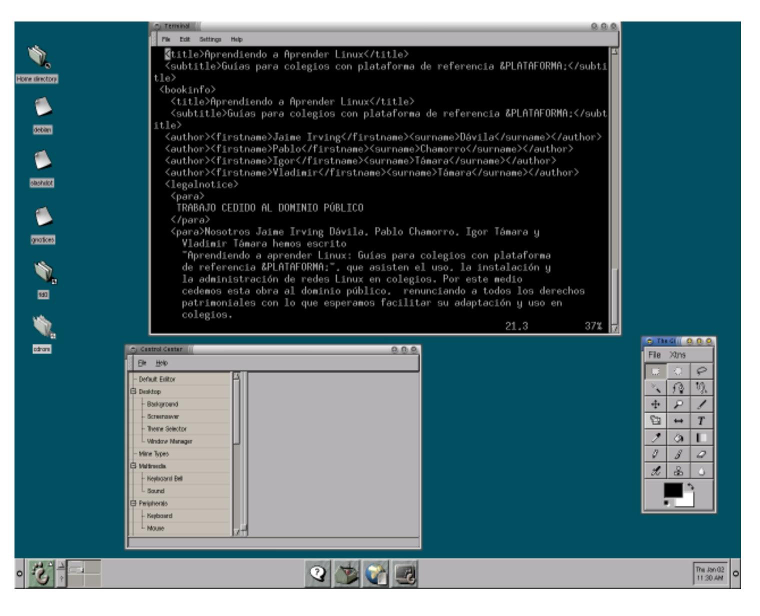
Se encuentra disponible actualmente en 48 idiomas en su última versión.

Historia de GNOME

El proyecto GNOME (GNU Network Object Model Environment) surgió en agosto de 1997 como proyecto liderado por los mexicanos Miguel de Icaza y Federico Mena para crear un entorno de escritorio completamente libre para sistemas operativos libres, en especial para GNU/Linux. Desde el principio, el objetivo principal de GNOME ha sido proporcionar un conjunto de aplicaciones amigables y un escritorio fácil de utilizar. GNOME también es una palabra del idioma inglés que significa gnomo.

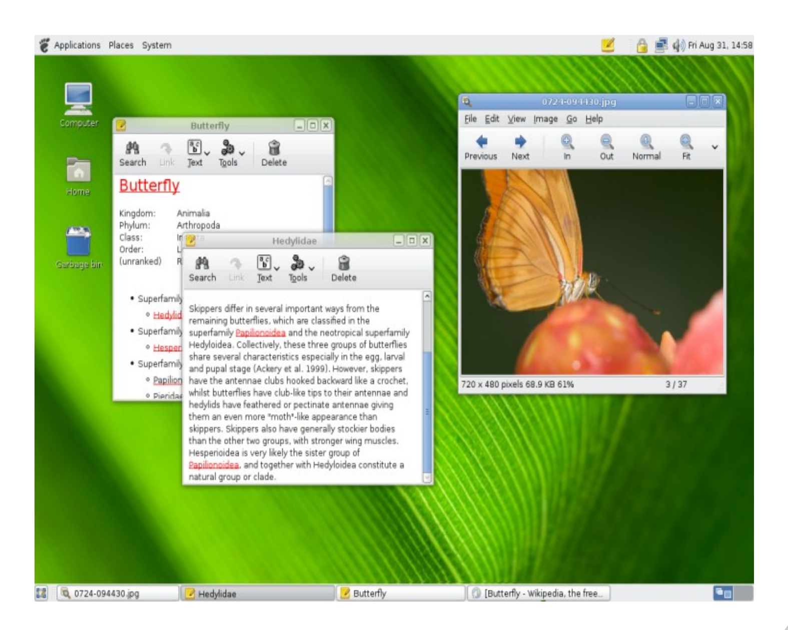
En esos momentos existía otro proyecto anterior con los mismos objetivos, pero con diferentes medios: KDE. Los primeros desarrolladores de GNOME criticaban a dicho proyecto por basarse en la biblioteca de controles gráficos Qt, cuya licencia (QPL), aunque libre, no era compatible inicialmente con la licencia GPL de la FSF.

Años más tarde los problemas de licencia de Qt se han resuelto y estas críticas han cesado. Sin embargo, los dos proyectos siguen rumbos tecnológicos distintos y se hacen una competencia amigable.



A medida que el proyecto ha ido progresando en los últimos años, los objetivos del mismo se han extendido para tratar una serie de problemas en la infraestructura Unix existente.

Actualmente el proyecto evoluciona bajo amparo de la Fundación GNOME.





Xfce

Es un entorno de escritorio ligero para sistemas tipo Unix como Linux, BSD, Solaris y derivados. Se configura íntegramente con el ratón o *mouse*. Su creador, Olivier Fourdan, dice de él: «Diseñado para la productividad, las aplicaciones se cargan y se ejecutan rápidamente, mientras conserva recursos de sistema».

El nombre Xfce originalmente provenía de **XForms Common Enviroment**, pero debido a los grandes cambios en el código, ya no usa el kit de herramientas de XForms, como originalmente lo hacía. El nombre sobrevivió, pero ya no se indica como XFce sino Xfce. Los desarrolladores están de acuerdo en que el nombre carece de significado actualmente, aunque se le suele desglosar como **X Free Choresterol Environment** (entorno X libre de colesterol) en referencia al poco consumo de memoria que realiza y a la velocidad con que se ejecuta al no tener elementos superfluos a diferencia de otros entornos de escritorio más grandes.

Xfce también provee el marco de trabajo para el desarrollo de aplicaciones. Además de Xfce mismo, hay otros programas que también utilizan las bibliotecas de Xfce, como el editor de texto Mousepad, el reproductor multimedia Xfmedia o el emulador de consola Terminal.

Xfce está basado en la biblioteca GTK+ 2.x y utiliza el gestor de ventanas Xfwm. Xfce se parecía en sus inicios al entorno de escritorio CDE, pero fue alejándose notablemente debido a que fue reprogramado nuevamente desde cero (ya lo había hecho entre las versiones 2.x y 3.x), y a diferencia de sus anteriores versiones, ahora cuenta con un sistema modular pudiendo gestionar un sistema de tipo multihead de manera bastante sencilla, y sigue todos los estándares establecidos por Freedesktop.org.

Thunar es el nuevo gestor de archivos predeterminado para Xfce desde la versión 4.4. Es similar a Nautilus y está diseñado para una máxima velocidad y un mínimo consumo de memoria. Xfce también posee un gestor de archivos comprimidos llamado Xarchiver.

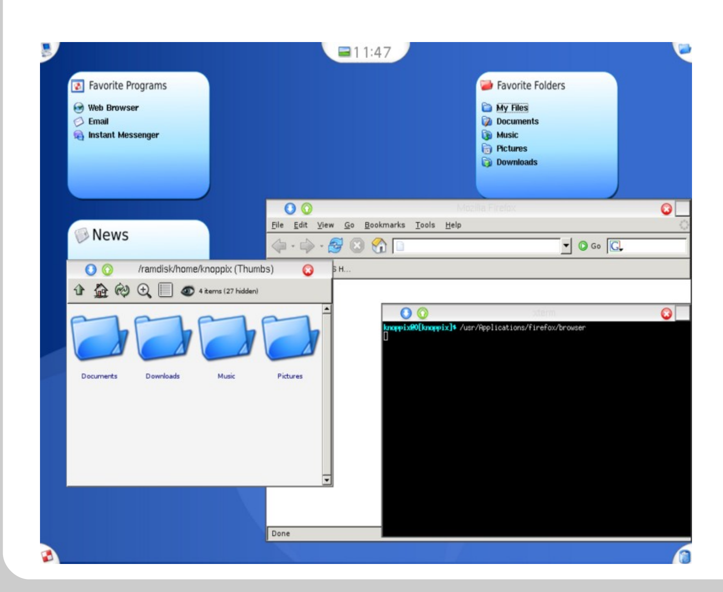


Mezzo

Mezzo es un entorno de escritorio creado por Jason Spisak.
Adicionado a la distribución linux llamada Symphony OS, que posee una nueva forma de presentar los datos para el usuario. Mezzo usa FVWM como gestor de ventanas.

Originalmente sólo estaba disponible para la distribución linux Symphony OS, pero ahora esta disponible para otras distribuciones. El diseño de Mezzo tiene también influencias de otros proyectos, particularmente del escritorio Kuartet, el cuál está construido sobre KDE usando Superkaramba y Python.

Mezzo se deshace de conceptos estándar como "El escritorio en un carpeta" y sistemas de menús jerarquizados, presentando toda la información necesaria directamente al usuario, desde el escritorio principal, y cuatro objetivos para las tareas y archivos relacionados con el Sistema, Programas, Archivos y Papelera.





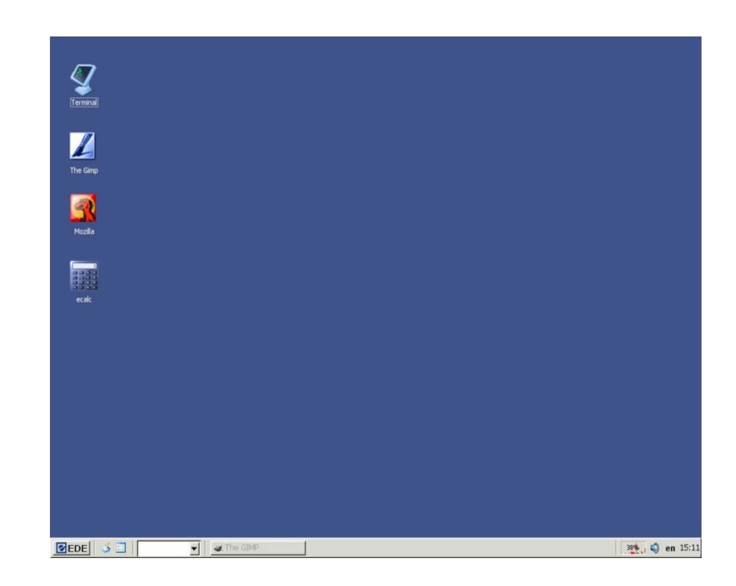
EDE

Equinox Desktop Environment es un pequeño ambiente de escritorio que es sinonimo de simpleza y rapidez.

Esta basado en una versión modificada de FLTK llamado eFLTK, sin embargo EDE 2 estará basado en la versión normal de FLTK.

Este escritorio tiene una apariencia y comportamiento similar a la apariencia clásica de Microsoft Windows. Además esta diseñado para ser muy ligero en el uso de la memoria y de los recursos del equipo.

Si tienes un equipo antiguo, este escritorio puede ser tu alternativa. Tener un hardware muy anticuado, no significa que no puedas colocar un escritorio moderno.





Enlightenment

Enlightenment, también conocido simplemente como E, es un gestor de ventanas ligero para UNIX y GNU/Linux. Uno de sus objetivos es llegar a ser un entorno de escritorio completo. Es muy configurable y visualmente es muy atractivo. Durante un tiempo fue el gestor de ventanas de GNOME.

La última versión estable es la 0.16.8.6 (también llamada DR16). El siguiente lanzamiento importante será la versión 0.17 (DR17) que está actualmente en fase de desarrollo y se basa en las nuevas Enlightenment Foundation Libraries (EFL). DR17 no está basado en DR16 sino que ha sido reescrito totalmente.



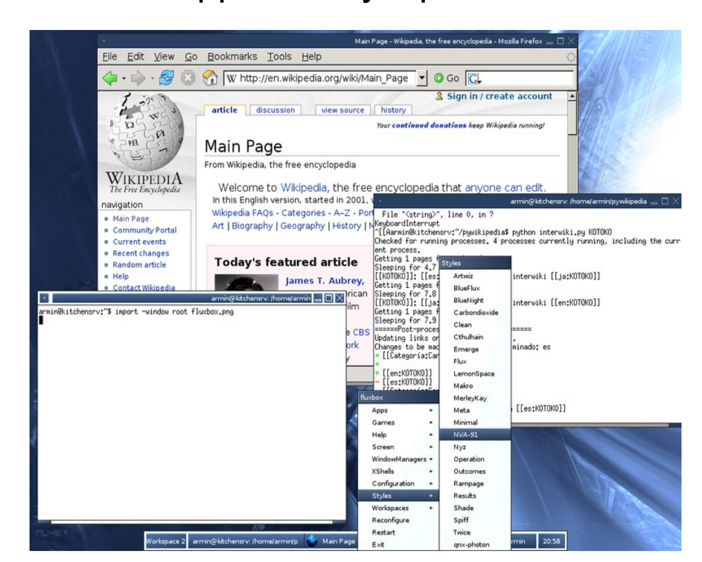


Fluxbox

Fluxbox es un gestor de ventanas para el Sistema X Window basado en Blackbox. Su objetivo es ser ligero y altamente personalizable, con sólo un soporte mínimo para iconos, gráficos, y sólo capacidades básicas de estilo para la interfaz. Algunos usuarios prefieren Fluxbox sobre otros gestores de ventanas debido a su velocidad y simplicidad.

Se puede utilizar iDesk, fbdesk ó ROX Desktop, en conjunto con Fluxbox, y para mejoras adicionales gadgetry.

Fluxbox es el Gestor de ventanas por defecto utilizado en Damn Small Linux debido a su pequeño consumo en memoria y un rápido tiempo de carga, Fluxbox es popular en muchos LiveCDs como Knoppix STD y Gparted.



CONCLUSIÓN

KDE es muy vistoso y flexible, es altamente personalizable, hay muchas aplicaciones desarrolladas para este escritorio. Como desventaja podríamos indicar que es un poco exigente en los recursos del equipo.

Mención aparte merece KDE4, un escritorio moderno y vistoso, estoy seguro que lo disfrutarás, porque es como darle un nuevo aire a la distro en donde lo utilices.

GNOME es un entorno muy rápido y eficiente, no consume tantos recursos como KDE, se puede personalizar, pero nunca al nivel que ofrece KDE. Se destaca su integracion con Xgl.

XFCE es uno de los entornos mas recientes que se viene incorporando al mundo Linux, su uso se ha masificado, es muy ligero y bonito, pero mas dificil de personalizar. Es parecido a Gnome en cuanto a estetica.

FLUXBOX es uno de los más rapido que existe, no tiene tantas características como los mencionados anteriormente, ni tampoco es muy personalizable, y es un poco mas complicado de usar, pero correrá perfecto en maquinas con muy pocos recursos.

MEZZO cambia el aspecto al reformular el concepto de escritorio, el cual se enfoca en proveer un acceso mucho más directo y funcional a las aplicaciones del sistema operativo, por lo menos de acuerdo a las leyes del diseño de interfaces de Jason Spisak's. Si deseas probar algo totalmente diferente, este es tu escritorio.

ENLIGHTMENT o mejor conocido como E17, es un ligero gestor de ventanas para UNIX y GNU/Linux. Uno de sus objetivos es llegar a ser un entorno de escritorio completo. Es muy configurable y visualmente es muy atractivo. Durante un tiempo fue el gestor de ventanas de GNOME.

EDE es un escritorio con una apariencia y comportamiento similar al escritorio clásico de Windows. Si tienes un equipo antiguo, este escritorio puede ser tu alternativa.

En resumen podemos indicar que no hay escritorios perfectos, cada uno tiene sus cualidades y desventajas. Sin embargo, gracias al software libre, puedes eligir el que mas te guste y el que mejor se adapte a tus necesidades.

Enlaces:

http://es.wikipedia.org/wiki/Common_Desktop_Environment
http://www.kde.org
http://www.gnome.org
http://www.xfce.org/
http://equinox-project.org
http://www.enlightenment.org/
http://www.fluxbox.org/
http://es.wikipedia.org/wiki/Mezzo_(entorno_de_escritorio)
http://en.wikipedia.org/wiki/FLTK

Alex Sandoval Morales
alexsandovalm@yahoo.com
http://www.microteknologias.cl
Miembro del equipo de Begins

trickle

Optimizando el uso de tu ancho de banda

Más de uno ha querido alguna vez dividir su ancho de banda para los diferentes programas que utiliza a diario, y sobre todo cuando tenemos una conexión de Internet de escaso ancho de banda.

Hay programas en los cuales podemos indicar cuánto pueden consumir de nuestra conexión (Wget, Azureus, etc..) pero son solo algunos. En las situaciones en que estos programas son ejecutados como superusuario (root) estos tienen una mayor prioridad y por lo tanto se pueden llevar toda la conexión; ejemplo de es al instalar algún software (a través de gestores de paquetes como apt-get, synaptic, aptitude, etc..) y tratar de navegar con Firefox: se nos hace casi imposible. Para esta problemática es que se creó Trickle, un software que trabaja a bajo nivel pero es muy fácil de utilizar.

En resumidas palabras Trickle es un software que permite manejar el ancho de banda a nivel de usuario. Se puede definir límites de subida y/o bajada a un determinado programa y funciona sobre una terminal.

Trickle tiene dos parámetros que te permiten limitar subida y/o bajada, estos son: -u y -d

Un ejemplo aplicando lo anterior podríamos fijar para que al instalar Apache2 aptitude nos ocupe hasta 10KB/s

```
trickle -d 10 aptitude install apache2
```

También existe la posibilidad de ejecutarlo como daemon y especificando nuestros limites. Ejemplo

```
trickled -d 20 -u 10
```

Luego podremos ejecutar

```
trickle mi_programa
trickle otro_programa
```

Con esto tendríamos a estos 2 programas consumiendo no más de 20 K/s de bajada y 10 K/s de subida.

Por último, tienes la disponibilidad que fijar los límites en tiempo real con tricklectl luego de haber ejecutado el daemon[1]

Y eso sería. Fácil y útil.

Autor: Marius Aamodt Eriksen
Licencia: BSD.
Sitio Web: http://monkey.org/~marius/trickle/
Última versión: 1.06 del 2003-06-02

1: Proceso que se ejecuta en segundo plano



Por Rodrigo Ramírez Norambuena
decipher@linuxchillan.cl

ZONA DE LINKS
EN ESTA EDICION: AUDIO



Amarok
http://amarok.kde.org/



EasyTag
easytag.sourceforge.net/



VLC
ww.videolan.org/vlc/



XMMS
www.xmms.org



Rythmbox
www.gnome.org/projects/rhythmbox/



Sound Juicer
http://burtonini.com/blog/computers/sound-juicer



Audacity
audacity.sourceforge.net



Sound Converter
http://soundconverter.berlios.de/



Banshee
http://banshee-project.org/



Audacious
http://audacious-media-player.org



Exaile
http://www.exaile.org/



Listen
http://www.listen-project.org/



Juk
http://developer.kde.org/~wheeler/juk.html



Begins



Envíanos tus artículos, tutoriales, entrevistas, sugerencias, etc.

articulos@revistabegins.org

http://revistabegins.org

http://www.linuxchillan.cl/?q=node/203

